HEWLETT-PACKARD

HP-75

BENUTZERHANDBUCH

HINWEIS

Hewlett-Packard übernimmt weder ausdrücklich noch stillschweigend irgendwelche Haftung für die in diesem Handbuch dargestellten Programm-Materialien – weder für ihre Funktionsfähigkeit noch für ihre Eignung für irgendeine spezielle Anwendung. Die Programm-Materialien dienen lediglich der beispielhaften Illustration; der Benutzer trägt das volle Risiko bezüglich ihrer Güte und Durchführbarkeit. Sollten sich Programm-Materialien als fehlerhaft erweisen, trägt der Benutzer (und nicht Hewlett-Packard oder irgendeine dritte Seite) die Kosten für die Korrekturen und alle Neben- und Folgeschäden. Hewlett-Packard haftet für keinerlei Neben- oder Folgeschäden im Zusammenhang mit oder als Folge aus der Lieferung, Benutzung oder Leistung der Programm-Materialien.

HEWLETT PACKARD

HP - 75
Benutzerhandbuch

October 1982

00075-90006

Printed in Singapore

Vorwort

Als erster Rechner einer neuen Generation von tragbaren Hewlett-Packard Computern verknüpft der HP-75C die Leistungsfähigkeit eines Hochsprachen-Computers mit den Vorteilen eines tragbaren Computersystems. Der HP-75C verfügt über 16K (16384) Bytes an Schreib/Lese-Speicher (RAM - Random Access Memory), die vom Benutzer zur Speicherung von Programmen, Texten und Daten verwendet werden können. Zusätzlich enthält der Rechner einen Festwertspeicher (ROM – Read Only Memory) von 48K Bytes, der dem Betriebssystem zugewiesen ist.

Aufbau dieses Handbuchs

Dieses Handbuch wurde als *Lern-* und *Nachschlagewerk* angelegt. Zuerst werden Sie mit seiner Hilfe lernen, wie Ihr Computer funktioniert, und mit allen Fähigkeiten des HP-75 vertraut gemacht werden. Obwohl bei der Konzeption des HP-75 Wert auf einfache Bedienbarkeit gelegt wurde, werden Sie dennoch einige Zeit in diesen Lernprozeß investieren müssen, um seine Vielseitigkeit und ausgefeilte Technik sinnvoll nutzen zu können. Nachdem Sie gelernt haben, Ihren Computer zu benutzen, wird Ihnen das Handbuch als umfassendes Nachschlagewerk für jedes Detail der Bedienung Ihres Computers dienen.

Teil I des Handbuchs, «Einführung», macht Sie mit dem HP-75 bekannt und gibt Ihnen Anleitungen zur ersten Benutzung des Computers. Lesen Sie zuerst Teil I – Sie sollten in wenigen Stunden mit den Anweisungen vertraut sein und die Beispiele durcharbeiten können. Am Ende von Teil I sind Sie in der Lage, den HP-75 ein- und auszuschalten, ein aufgezeichnetes BASIC-Programm einzulesen und auszuführen, die Anzeige mit dem Tastenfeld zu kontrollieren, und Text- und BASIC-Files im Speicher zu editieren und zu bearbeiten.

Teil II, «Gebrauch des HP-75», beschreibt alle wesentlichen Eigenschaften des HP-75 im Detail. Einige Abschnitte in Teil II beschreiben Möglichkeiten, die Sie vielleicht im Moment nicht anwenden wollen. Überspringen Sie einfach diese Abschnitte, und lesen Sie nur, was Sie benötigen. Am Ende von Teil II können Sie Tastenfeldberechnungen mit Operatoren und Funktionen durchführen , die Systemuhr benutzen, stellen und justieren, Termine planen und bestätigen, Kartenleseroperationen durchführen, das eingebaute HP-IL Interface benutzen und das Tastenfeld umdefinieren.

In Teil III, «Programmierung», wird die Programmierung des HP-75 in BASIC erläutert. Teil III setzt gewisse Programmiererfahrung in BASIC oder einer anderen höheren Programmiersprache voraus. Wenn Sie noch nie programmiert haben, ist es zu empfehlen, ein Lehrbuch zu den Grundlagen der Programmierung in BASIC durchzuarbeiten, bevor Sie mit Teil III beginnen. Am Ende von Teil III können Sie BASIC-Programme schreiben, editieren, starten und korrigieren und alle Systembefehle und BASIC-Anweisungen, über die der HP-75 verfügt, einsetzen.

Die Anhänge am Ende dieses Handbuches sind ein wertvolles Nachschlagwerk. In Anhang G, Glossar, werden viele der in diesem Handbuch benutzten Ausdrücke definiert, – es empfiehlt sich, beim Durcharbeiten dieses Handbuchs schon früh das Glossar zu Rate zu ziehen. In den Anhängen C, D, E und H befinden sich die wichtigsten Zusammenfassungen zum Betrieb und zur Programmierung des Computers. Anhang B enthält Informationen zur Pflege des Computers und zu Service-Einrichtungen, und in Anhang A finden Sie eine Auflistung des Standardzubehörs, das mit Ihrem Computer mitgeliefert wurde. Anhang F schließlich enthält die Listings aller Programme in Ihrem Benutzerpaket.

Dieses Handbuch enthält zwei Sachregister: ein umfassendes thematisches Register und, auf der Innenseite des Rückumschlags, ein vollständiges Register der Anweisungen des HP-75. Mit dem Anweisungsregister können Sie die Beschreibungen aller Befehle, Anweisungen, Funktionen und Operatoren Ihres Computers schnell auffinden.

Das *HP-75 Referenzhandbuch,* das Ihrem HP-75 beiliegt, enthält die Anweisungen und Operationen des Computers in komprimierter Form und ist handlich genug, um den HP-75 überall hin zu begleiten. Wenn Sie Teil I beendet haben und die Teile II und III durcharbeiten, kann Ihnen das Referenzhandbuch als kompaktes Nachschlagewerk dienen.

Das Computersystem HP-75

Der HP-75 wurde als Kontrollgerät eines mobilen Computer*systems* entworfen. Das eingebaute HP-IL Interface gibt Ihnen die Möglichkeit, eine Vielzahl von Peripheriegeräten an den HP-75 anzuschließen und dadurch seine Leistungsfähigkeit weiter zu erhöhen. In die drei Einschubschächte an der Vorderkante des HP-75 können Sie Module mit weiteren Befehlen und BASIC Anweisungen oder einer Vielzahl von Anwendungsprogrammen einstecken. Lesen Sie den Zubehörkatalog, der Ihrem Computer beiliegt, um alles über die Peripheriegeräte und Softwareprodukte von Hewlett-Packard zu erfahren.

Inhalt

Teil III: Programmierung des HP-75

Anhänge

Register

Das Tastenfeld des HP-75

Systemtasten sind schwarz, Editierungstasten sind schattiert dargestellt.

Masken des HP-75

In diesem Handbuch wird ein blau schattiertes Rechteck verwendet, um anzudeuten, daß in der Anzeige des HP-75 der Cursor über einem Zeichen erscheint.

TIME-Modus

`SAT 02/05/1983 09:30:15 AM`	Typische TIME-Anzeige.
`Set Mo/Dy/Year Hr.Mn:Sc AM`	Zeitsetzmaske, die zum Stellen der Uhr benutzt wird.
`Date: MDY, ~Time: AM, Appt: YEAR`	STATS-Maske zur Eingabe des Zeit- und Datumsformats und zur Änderung des Terminbereichs.
`Adjust (N) +Hr+Mn+Sc,t,`	ADJST-Maske zur Justierung der Uhr.

APPT-Modus

`Day Mo/Dy/Yr Hr:Mn: AM #1N !Note`	APPT-Maske zum Einrichten von Terminen.
`Rept=Mo+Dy+Hr+Mn : DOW`	Wiederholungsmaske zur Wiederholung von Dauerterminen.
`Year? YYYY`	Jahresmaske zum Einrichten von Terminen im erweiterten Kalender.

EDIT-Modus

```
>
```

Anzeige im EDIT-Modus mit BASIC-Eingabeaufforderung und Cursor.

```
:
```

Anzeige im EDIT-Modus mit Texteingabeaufforderung und Cursor.

```
Name   Type Len   Time   Date
```

Erste Zeile eines CAT ALL Listings.

```
KONTEN   B 1242 10:20 02/21/83
```

Typischer Katalogeintrag eines Files.

```
>DEF KEY 'a','a;
```

Voreinstellung einer Tastendefinition.

```
>20 DIM A$[200],L$[96],T(6,20)
```

BASIC Programmanweisung.

Kartenleser

```
Catalog card: Align & [RTN]
```

Anzeige des Katalogeintrags einer Magnetkarte.

```
Copy to card: Align & [RTN]
```

Einlesen eines Files von einer Magnetkarte in den Speicher.

```
Verify card: Align & [RTN]
```

Zur Prüfung der Korrektheit einer Informationsübertragung auf eine Magnetkarte.

```
Copy from card: Align & [RTN]
```

Kopieren von Information vom Speicher auf Magnetkarte.

```
Protect card: Align & [RTN]
```

Zum Schutz von Magnetkarten gegen Überschreiben.

```
Unprotect card: Align & [RTN]
```

Zum Entfernen des Schreibschutzes.

Hewlett-Packard Interfaceschleife

```
Device # 1=':'
```

Wird vor Beginn von HP-IL Zuordnungen angezeigt.

Teil I
Ein Überblick

Abschnitt 1

Der Einstieg

Inhalt

Einführung

Sie haben mit Ihrem HP-75 einen sehr leistungsfähigen und vielseitigen Computer erworben. Bevor Sie den HP-75 benutzen, sollten Sie sich mit einigen seiner Eigenschaften vertraut machen.

Tastenfeld

Sie stehen über das 65-Tasten Tastenfeld mit dem Computer HP-75 in Verbindung. Wenn Sie eine falsche Taste drücken, können Sie eine Fehlermeldung auf der Anzeige erzeugen oder möglicherweise sogar den Inhalt des Speichers löschen, aber Sie können den HP-75 nicht durch das Drücken von Tasten beschädigen. Die Anzahl der möglichen Eingaben wird durch Tastenkombinationen vervielfacht.

AC-Adapter/Ladegerät

Bei Erhalt Ihres HP-75 ist der wiederaufladbare Batteriesatz bereits in den Computer eingebaut. Der Batteriesatz kann zu Beginn jedoch möglicherweise entladen sein. Ist dies der Fall, können Sie trotzdem sofort mit dem HP-75 zu arbeiten beginnen: Schließen Sie ihn einfach an das beigelegte Ladegerät an. Beachten Sie beim Anschließen des AC-Adapter/Ladegeräts:

> **Hinweis:** Beim Anschließen und Entfernen des AC-Adapter/Ladegeräts an den Computer sollten Sie den Computer ausgeschaltet haben. Ist die Anzeige des HP-75 eingeschaltet, schlagen Sie unter «Ein- und Ausschalten des HP-75» (Seite 13) nach.

1. Stecken Sie den Adapterstecker in die Adapterbuchse auf der Rückseite des HP-75.

2. Stecken Sie den Netzstecker des AC-Adapter/Ladegeräts in eine Netzsteckdose.

Diese zwei Schritte können auch in der umgekehrten Reihenfolge erfolgen. Sie können das Ladegerät auch jederzeit entfernen, wenn Sie das Gerät mit dem Batteriesatz betreiben wollen. Es ist zwar möglich, den HP-75 mit *ausgebautem* Batteriesatz über das Netz zu versorgen; dabei kann aber bei Spannungsschwankungen oder -unterbrechungen der Inhalt des Speichers verloren gehen.

Wenn Sie den HP-75 über das Netz benutzen, wird das Ladegerät und der Deckel des Batteriefachs warm; dies ist kein Grund zur Beunruhigung.

Einschalten des HP-75 (ATTN)

Drücken Sie bei geladener Batterie oder eingestecktem Netzadapter die Taste ATTN (*Attention*), um den Rechner einzuschalten. Sie sehen eine von zwei möglichen Anzeigen:

```
Set Mo/Dy/Year Hr:Mn:Sc AM
```

Zeitsetz-Maske.

oder

```
>
```

Cursor des *Edit*-Modus.

Wenn die erste Anzeige erscheint, müssen Sie die System-Uhr stellen. Lesen Sie weiter unten in diesem Abschnitt «Stellen der System-Uhr».

Wenn die zweite Anzeige erscheint, müssen Sie die Zeitanzeige überprüfen und gegebenenfalls neu stellen. Drücken Sie dazu die Modustaste [TIME] oben links auf dem Tastenfeld. Sie sehen eine Anzeige wie diese:

```
SAT 02/05/83 09:18:17 AMu
```

Zeit-Anzeige.

Natürlich werden die Zeit und das Datum von den in diesem Beispiel gegebenen verschieden sein. Wenn das richtige Datum und die richtige Uhrzeit angezeigt werden, brauchen Sie die weiteren Schritte dieses Abschnitts nicht durchzuführen. Wenn Zeit und Datum jedoch *nicht* richtig sind, tasten Sie das Wort «set» ein und drücken Sie die Taste [RTN] (*Return*). Dann sehen Sie die erste der oben gezeigten Anzeigen – die Zeitsetz-Maske:

```
Set Mo/dy/year Hr:Mn:Sc AM
```

Zeitsetz-Maske.

Einstellen der System-Uhr. Die Zeitsetz-Maske enthält alle Information, die Sie zum Stellen der System-Uhr benötigen. Der Ersetzungscursor, ▒ , ist ein kontinuierlich blinkendes Rechteck, das sich beim Eingeben in der Anzeige nach rechts bewegt. (In diesem Handbuch wird ein schattiertes Rechteck zur Darstellung des Cursorzeichens benutzt, wenn der Cursor über einem anderen Zeichen in der Anzeige steht.) Geben Sie mit Hilfe der Ziffernreihe auf dem Tastenfeld die Datums-und Zeitinformationen ein. In der obenstehenden Zeitsetz-Anzeige wird die erste Ziffer, die Sie eingeben, über das `M` in `Mo` (*Monat*) geschrieben.

Fall Sie einen Eingabefehler machen, drücken Sie einfach die Taste [BACK] (*Backspace*) oder [CLR] (*Clear*) und setzen die Eingabe fort. Mit [CLR] können Sie die Eingabe noch einmal von vorn beginnen. Geben Sie die folgenden Informationen in die Zeitanzeige ein:

`Mo` Laufender Monat. Geben Sie eine Zahl zwischen `01` (für Januar) und `12` (für Dezember) ein. Benutzen Sie zur Eingabe der Null die Taste [0] und nicht die Taste [O].*

`Dy` Tag des Monats. Geben Sie eine Zahl zwischen `01` und `31` ein.

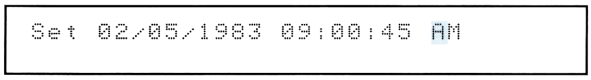
`Year` Jahr, eine vierstellige Zahl.

`Hr` Stunde, eine Zahl zwischen `01` und `12`.

`Mn` Minute, eine Zahl zwischen `00` und `59`.

`Sc` Sekunde, eine Zahl zwischen `00` und `59`. Setzen Sie die Sekunden auf einen Zeitpunkt, der noch ca. 45 Sekunden in der Zukunft liegt.

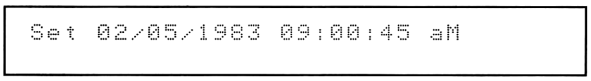
* In den Feldern Monat, Tag, Stunde, Minuten und Sekunden können Sie anstelle der Null auch ein Leerzeichen verwenden. `06`, *Leerzeichen* `6` und `6` *Leerzeichen* im Monatsfeld bezeichnen alle den Monat Juni.



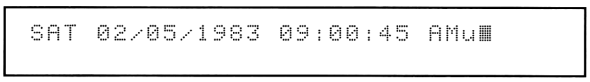
Die Anzeige nach der Eingabe obenstehender Informationen. In den Beispielen beginnt «heute» am Samstag, dem 5. Februar 1983, um 9 Uhr.

Geben Sie ein a für **morgens** (p für **nachmitags**) ein. Drücken Sie dann zur angegebenen Zeit die Taste RTN (*return*). Die Taste RTN stellt die Uhr innerhalb 0,1 Sekunden auf die angezeigte Zeit.

Hinweis: Wenn in diesem Handbuch zwei Anzeigeillustrationen wie die folgenden nebeneinandergestellt sind, werden durchgehend im linken Feld die einzugebenden Informationen gezeigt, und das rechte Feld zeigt das Ergebnis nach dem Drücken der Taste RTN.



Die Vollständige Maske.



Der Wochentag wird automatisch berechnet, und die Zeitanzeige wird jede Sekunde aktualisiert.

Der HP-75 benutzt die System-Uhr bei vielen Operationen, unter anderen bei der Verwaltung von Terminen, bei der Datierung von Files und bei der Ausführung von Programmen.

Ausschalten des HP-75 (SHIFT ATTN)

Die Taste ATTN schaltet den HP-75 ein; sie aktiviert die Anzeige und das Tastenfeld.

Halten Sie, um den Computer *auszuschalten*, die Taste SHIFT gedrückt, und drücken Sie zusätzlich die Taste ATTN. Damit werden die Anzeige und das Tastenfeld desaktiviert.

Der HP-75 benötigt eine kontinuierliche Stromversorgung zur Erhaltung der Speicherinhalte und zum Betrieb der Uhr. Der Stromverbrauch ist aber bei aktiviertem Computer (Anzeige in Betrieb) höher als bei inaktivem Computer (Anzeige ausgeschaltet). Der Computer schaltet sich, um Strom zu sparen, selbst *aus*, nachdem er für etwa fünf Minuten keine Eingabe mehr erhalten hat. Unabhängig davon, ob Sie den Computer ausgeschaltet haben oder er sich selbst, können Sie ihn mit der Taste ATTN wieder einschalten.

Zurücksetzen des HP-75 (SHIFT CTL CLR)

Beim *Zurücksetzen* geht der gesamte Speicherinhalt verloren, der Computer ist im Einschaltzustand nach dem allerersten Einsetzen des Batteriesatzes. Sie können den HP-75 auf drei Weisen zurücksetzen:

- Vollständiges Entladen des Batteriesatzes.
- Herausnehmen des Batteriesatzes bei eingeschalteter Anzeige, während der Computer nicht ans Ladegerät angeschlossen ist. Die RAM-Schaltkreise werden sofort entladen.
- Zurücksetzen mit einer Dreitasten-Folge vom Tastenfeld aus. Halten Sie die beiden Tasten SHIFT *und* CTL gedrückt, und drücken Sie zusätzlich CLR für ungefähr eine Sekunde.*

Wenn Sie das Gerät je zurücksetzen wollen, müssen Sie danach den Einschaltvorgang von Anfang an mit der Taste ATTN und dem Stellen der Uhr beginnen. Probieren Sie jetzt die Tastenfolge SHIFT CTL CLR, wenn Sie wollen, aber in Zukunft sollte das nicht notwendig sein.

Um den Batteriesatz zu entfernen, ohne den Computer zurückzusetzen müssen Sie den HP-75 zuerst ausschalten (mit SHIFT ATTN), bevor Sie das Batteriefach öffnen. *Drücken Sie nicht* ATTN, solange die Batterie nicht im Batteriefach ist. Wenn das Gerät nicht an eine Steckdose angeschlossen ist, sollten Sie die Batterie nicht länger als für 30 Sekunden aus dem Batteriefach entfernen.

Schlagen Sie zusätzliche Informationen zum Thema Batterien in Anhang B nach.

---

* Zusätzlich zu SHIFT CTL CLR bewirken auch die folgenden Tastenfolgen ein Zurücksetzen des Systems, wenn sie für mehr als eine Sekunde gedrückt bleiben: SHIFT CTL gefolgt von I/R, 8, I, J oder M.

Die Tasten [ATTN] und [RTN]

Die Taste [ATTN] wurde aus gutem Grund in die obere linke Ecke gesetzt – sie ist von großer Wichtigkeit. Sie wird zur Unterbrechung fast aller Systemoperationen und zur Aktivierung des HP-75 für die Tastenfeldeingabe benötigt.

Die Taste [ATTN] aktiviert den HP-75; und die Taste [RTN] bringt ihn zum Arbeiten. Wenn Sie [RTN] drücken, erfolgt eine oder mehrere der folgenden Reaktionen:

- Der Inhalt der Anzeige wird wie beim Stellen der Uhr abgespeichert.
- Der soeben eingegebene Befehl oder die eingegebene Anweisung wird ausgeführt.
- Der HP-75 informiert Sie durch Tonsignal und Blinken über aufgetretene Fehler. Sie können den Computer nach den meisten Fehlermeldungen mit der Taste [ATTN] wieder in Betriebsbereitschaft versetzen.

Sobald Sie wissen, wann Sie die Taste [RTN] drücken müssen – und wann nicht – können Sie den HP-75 kontrollieren. Dies sollten Sie am Ende des Abschnitts 3 «Editieren von Files» gelernt haben.

Das Anzeigefenster

Das Anzeigefenster zeigt 32 Zeichen gleichzeitig an. Jede Anzeige*zeile* kann bis zu 96 Zeichen einschließlich Cursor enthalten. Wenn Sie mit einer Eingabe das gesamte 32 Zeichen-Anzeigefenster gefüllt haben, wird bei der Eingabe weiterer Zeichen die Information nach links aus der Anzeige geschoben.

Das Anzeigefenster enthält vier *Statusanzeigen*, die Sie über besondere Betriebsbedingungen informieren.

BATT	ERROR	PRGM	APPT

Gewöhnlich unsichtbar, erscheinen diese nur bei folgenden Anlässen:

BATT: Die Batteriespannung fällt ab.

ERROR: Der HP-75 kann eine Ihrer Anweisungen nicht verstehen oder hat beim Programmablauf einen Fehler entdeckt.

PRGM: Es wird gerade ein Programm ausgeführt.

APPT: Ein fälliger Termin steht an.

Die drei Betriebsmodi

Wenn Sie eine der drei Tasten [TIME] (Zeit), [APPT] (Termin, Verabredung) oder [EDIT] (Korrektur) drücken, schaltet der HP-75 in den angegebenen *Modus* oder Betriebszustand.

Der TIME-Modus wird benötigt:

- Um die Systemuhr zu stellen.
- Um die Zeit anzuzeigen.
- Um neue Zeit- und Datumsformate zu spezifizieren.
- Um die Ganggeschwindigkeit der Uhr zu verändern.

Der APPT-Modus wird benutzt:

- Zur Planung persönlicher und geschäftlicher Termine.
- Zum Prüfen von Kalenderdaten.

Der EDIT-Modus – das «Arbeitspferd» der drei Modi – wird benötigt:

- Für Berechnungen über das Tastenfeld.
- Zum Schreiben und zur Ausführung von BASIC-Programmen.
- Zum Schreiben und Lesen von Informationen auf Magnetkarten.
- Zur Neudefinition von Tasten und Tastenfolgen für die Anzeige von Meldungen und die Ausführung von Befehlen.
- Zum Schreiben von Memos.
- Zur Steuerung von Druckern und anderen Peripheriegeräten über die Hewlett-Packard Interface Loop.

Drücken Sie nun nacheinander alle drei Tasten:

[TIME]

```
SAT 02/05/1983 09:15:47 AM
```

Schaltet den HP-75 in den TIME-Modus.

[APPT]

```
Day Mo/Dy/Yr Hr:Mn AM #1N !Note
```

Schaltet den HP-75 in den APPT-Modus.

[EDIT]

```
>
```

Schaltet den HP-75 in den EDIT-Modus.

In diesem Abschnitt werden die Fähigkeiten aller drei Modi – TIME, APPT und EDIT – kurz umrissen.

Der TIME-Modus ([TIME])

Wenn Sie die Taste TIME drücken, schaltet der HP-75 in den TIME-Modus.

Sie können die Uhr auf eine andere Zeit stellen, wenn Sie das Wort set in die TIME-Anzeige eintasten und die Taste [RTN] drücken:

```
SAT 02/05/83 09:18:17 AM set
```

```
Set Mo/Dy/Year Hr:Mn:Sc AM
```

Drücken Sie [ATTN], wenn Sie die Zeit nicht ändern wollen, oder geben Sie die neue Information ein und drücken Sie [RTN]. Wenn Sie einen Teil der Information auslassen (wie Monat, Tag oder Jahr) wird dieser von der momentanen Systemzeit übernommen.

Der Befehl SET ist einer von fünf Befehlen des TIME-Modus. In Abschnitt 6 «TIME-Modus Operationen» wird die Anwendung dieser Befehle bei der Spezifikation neuer Anzeigeformate und beim Stellen der Uhr erläutert. In den Beispielen dieses Handbuchs werden gewöhnlich die Formate Monat/Tag/Jahr und AM/PM verwendet.

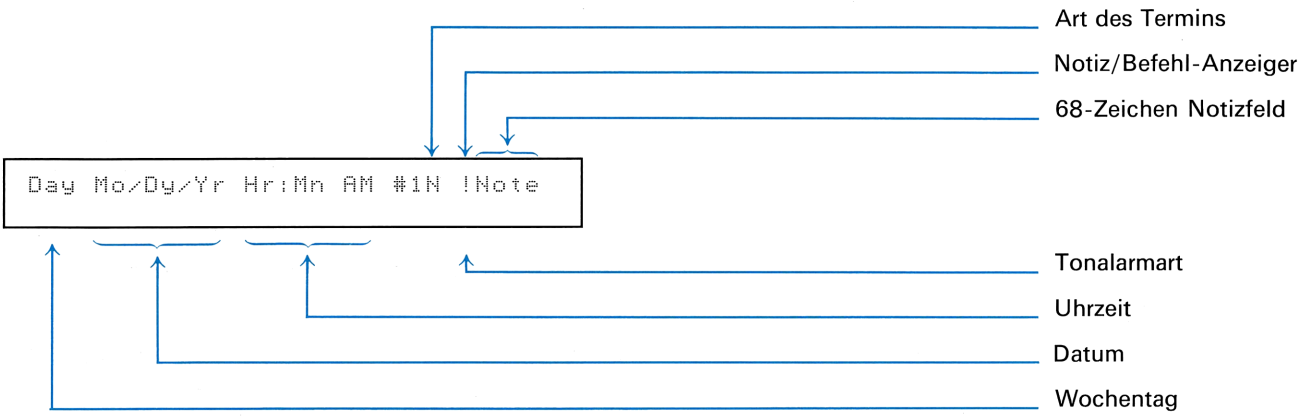
> **Hinweis:** Während Sie dieses Handbuch lesen, könnte die Anzeige verschwinden – erinnern Sie sich, daß sich der HP-75 nach einiger Zeit selbst ausschaltet. Schalten Sie ihn mit [ATTN] wieder ein. Beim Einschalten befindet sich der Computer immer im EDIT-Modus.

Der APPT-Modus (APPT)

Der HP-75 ist mit zwei Kalendern, einem Jahreskalender und einem auf 10.000 Jahre vorausberechneten erweiterten Kalender, ausgerüstet, um Ihnen die Zeitplanung für Ihre Termin-Alarme, Meldungen und Programme zu ermöglichen. Der APPT-Modus verbindet die Ganggenauigkeit der Uhr des HP-75 mit den Fähigkeiten des HP-75 Mikroprozessors. Sie können mit dem HP-75C mehr als 3000 verschiedene Termine planen, und jeder Terminaufruf kann nach Ablauf von Zeitspannen zwischen einer Minute und acht Jahren wiederholt werden.

Eingabe eines Termins (TAB, BACK, ←, →, RTN)

Schalten Sie den HP-75 zur Planung eines Termins in den APPT-Modus: Drücken Sie APPT. Die APPT-Anzeige erscheint:



Diese *Felder* oder Anzeigeteile werden in Abschnitt 7, «APPT-Modus Operationen» im Detail besprochen. Hier sind vorläufig nur die Tasten aufgelistet, die Sie zur Eintragung eines Termins benötigen:

- Die Taste TAB bewegt den Cursor auf dem Anzeigefeld nach rechts. Mit SHIFT TAB können Sie den Cursor nach links bewegen.
- Die Tasten BACK, ← (*Linkspfeil*) und → (*Rechtspfeil*) ermöglichen Ihnen Korrekturen der Eingabe.
- Die Taste CLR löscht die Anzeige und bringt die APPT-Maske wieder zurück.

Drücken Sie für dieses Beispiel die Taste TAB sechsmal um zur `1` im Alarm-Feld zu springen. Drücken Sie dann 3, um ein Zweiton-Alarmsignal zu setzen:

```
Day Mo/Dy/Yr Hr:Mn AM #3N !Note
```

Die APPT-Maske, nachdem die Alarmart geändert wurde. Zehn Alarmsignale stehen zur Verfügung (Seite 101).

Bewegen Sie den Cursor durch ein weiteres Drücken von TAB auf das Feld `Note`. Wenn der Termin fällig ist, wird Ihre Notiz angezeigt. Wir wollen eine kurze Notiz benutzen: `Anruf Meier: 8976631.`

```
Mn AM #3N !Anruf Meier: 8976631
```

Ihre Terminmeldung kann bis zu 68 Zeichen enthalten, das sind mehr als zwei volle Fenster.

Mit den Tasten ← und → können Sie die gesamte Terminmeldung wieder betrachten. Ist die Meldung nach Ihren Wünschen eingegeben, speichern Sie sie mit RTN ab. Die vollständige Meldung wird wiederholt:

```
SAT 02/05/83 09:25 AM #3N !Anruf
```

Der Beispieltermin ist so gewählt, daß er am Ende der nächsten Minute fällig wird.

Sie brauchen die APPT-Maske nicht vollständig zu füllen, da der HP-75 mit Hilfe des derzeitigen Datums, der Zeit und der eingegebenen Werte die Werte für die freigelassenen Felder selbst berechnet. Beachten Sie, daß der Cursor das Jahresfeld von vornherein überspringt; der HP-75 ist auf das Kalenderjahr gestellt, und Eingabe der Jahresinformation ist für Termine innerhalb des Kalenderjahres nicht notwendig.

Drücken Sie [CLR], wenn Sie die ungefüllte APPT-Maske anzeigen wollen.

Fälligwerden eines Termins ([APPT], [ATTN])

Gleich ob Sie im TIME-, APPT- oder EDIT-Modus sind, sobald ein Termin fällig wird, erklingt der Alarm, und die APPT-Statusanzeige wird eingeschaltet. Drücken Sie in diesem Beispiel die Taste [TIME], um die Zeit bis zum Eintreffen des Termins zu überwachen. Wenn der Sekundenzähler 00 erreicht, sehen Sie

```
SAT 02/05/1983 09:25:00 AM
                          APPT
```

In TIME-Modus wird ein Termin fällig, das Alarmsignal erklingt, und die APPT-Statusanzeige wird eingeschaltet.

Mit [APPT] können Sie den fälligen Termin anzeigen:

```
SAT 02/05/83 09:25 AM #3N !Anruf
                          APPT
```

Wochentag, Datum und Uhrzeit sind unterstrichen, um anzuzeigen, daß es sich um einen fälligen Termin handelt.

Mit [ATTN] bestätigen Sie die Terminmeldung und schalten die APPT- Statusanzeige aus. Mit den Tasten [↑] (*Aufwärtspfeil*) und [↓] (*Abwärtspfeil*) können Sie frühere und spätere Termine überprüfen; diese werden in der Reihenfolge ihrer Eingabe gespeichert.

Sollte ein Termin fällig werden, wenn der HP-75 ausgeschaltet ist, läßt der HP-75 den Alarm erklingen, schaltet den EDIT-Modus ein und zeigt die Notiz an. Schalten Sie das Gerät mit [SHIFT] [ATTN] aus, um dies zu probieren.

```
!Anruf Meier: 8976631
```

Im ausgeschalteten Zustand *bearbeitet* der HP-75 fällige und überfällige Termine. In diesem Fall zeigt er die Terminnotiz an.

Die Notiz bleibt für fünf Sekunden, oder bis Sie eine andere Taste drücken, in der Anzeige; danach schaltet sich der HP-75 wieder aus. Wenn Sie das Gerät wieder einschalten (mit [ATTN]) erscheint die APPT Statusanzeige, falls irgendwelche Termine bestätigt werden müssen.

Termine können nicht nur angezeigt werden, sondern auch zum Start von Programmen und zur Ausführung von anderen Programmanweisungen eingesetzt werden (siehe Abschnitt 7).

Der EDIT-Modus ([EDIT])

Der EDIT-Modus ist das «Kraftwerk» des HP-75. Praktisch alle Operationen des HP-75 werden im EDIT-Modus ausgeführt. Der Computer schaltet beim Einschalten des Geräts und wenn Sie die Taste [EDIT] drücken in den EDIT-Modus.

```
>
```

Die Anzeige im EDIT-Modus.

Die BASIC-Eingabeaufforderung, das Zeichen >, erscheint links in der Anzeige und sagt Ihnen, daß der HP-75 für den nächsten Befehl, die nächste Programmanweisung oder die nächste Tastenfeldberechnung bereitsteht. Der HP-75 verfügt über ein Zeichen zur Texteingabeaufforderung, den Doppelpunkt :, das Sie zur Eingabe von Memos und anderer Korrespondenz benutzen können.

Führen Sie folgende Schritte durch, um die BASIC-Eingabeaufforderung oder die Texteingabeaufforderung, falls nicht schon vorhanden, in der Anzeige zu erhalten:

1. Drücken Sie [EDIT].
2. Tasten Sie `purge` ein und drücken Sie [RTN]. Der Befehl PURGE löscht den momentanen Arbeitsspeicher des Computers.

3. Wählen Sie eine der folgenden Möglichkeiten:

   - Geben Sie `edit text` ein und drücken Sie [RTN], um die Texteingabeaufforderung (:) zu erhalten.

```
workfile    T     0 09:32 02/05/83
```

Die Anzeige enthält den Katalogeintrag eines temporären Text-Arbeitsfiles.

   - Geben Sie `edit basic` ein und drücken Sie [RTN], um die BASIC-Eingabeaufforderung zu erhalten:

```
workfile    B     0 09:32 02/05/83
```

Die Anzeige enthält den Katalogeintrag eines temporären BASIC-Arbeitsfiles.

   Der Zeitpunkt der Anlage des Files wird in 24-Stunden-Notation ausgedrückt. Eine 0 bedeutet, daß der File leer ist.

4. Drücken Sie [ATTN]. Die entsprechende Eingabeaufforderung erscheint:

```
:
```

Texteingabeaufforderung.

```
>
```

BASIC-Eingabeaufforderung.

Die Bedeutung dieser Schritte wird in Abschnitt 3 «Editieren von Files» deutlich.

Tastenfeld-Arithmetik ([+], SQR)

Im EDIT-Modus dient der HP-75 als leistungsstarker Rechner mit 64 eingebauten Funktionen und Operationen. Bei Tastenfeldoperationen muß die BASIC-Eingabeaufforderung (>) in der Anzeige stehen. (Schlagen Sie im vorhergehenden Abschnitt nach, wenn stattdessen die Texteingabeaufforderung in der Anzeige steht.) Um eine Reihe von Zahlen aufzuaddieren, geben Sie den Ausdruck ein und drücken Sie [RTN].

Beispiel:

```
>1+2+3+4+5+6
```

Das Gleichheitszeichen (=) wird bei der Tastenfeldarithmetik nicht benutzt.

```
21
```

Drücken der Taste [RTN] liefert das Ergebnis.

Der Punkt (.) dient als Dezimaltrennzeichen. Neben der Addition stehen die arithmetischen Operationen Subtraktion (-), Multiplikation (*), Division (/) und Exponentiation (^) zur Verfügung. Sie können jede Kombination von ganzzahligen und nicht ganzzahligen Werten mit diesen Operatoren berechnen. Der HP-75 arbeitet mit 12stelliger Genauigkeit. Schlagen Sie Details zur Tastenfeld-Arithmetik in Abschnitt 4 nach.

Geben Sie die folgende Tastenfolge ein, wenn Sie die Quadratwurzel von z.B. 78 berechnen wollen, und drücken Sie anschließend [RTN]:

```
>sqr(78)
```

Bei der Eingabe der Funktion SQR erscheint die Eingabe-Anforderung sofort nach dem Drücken der Taste [S] wieder.

```
8.83176086633
```

Das Ergebnis. Die letzte (zwölfte) Stelle wird gerundet.

Zahlen, Funktionen und Variablen können mit arithmetischen Operatoren (wie z.B. +), Vergleichsoperatoren (wie z.B. >) und logischen Operatoren (wie z.B. AND) zu komplexen Ausdrücken kombiniert werden. In Abschnitt 5 werden numerische Funktionen und Ausdrücke besprochen.

Eingaben im EDIT-Modus (' ' und " ")

Bei Vorhandensein der BASIC-Eingabeabfrage (>) behandelt der HP-75 Kleinbuchstaben bei der Eingabe genauso wie Großbuchstaben. Sie können also Ihre Befehle und Programme wahlweise in Groß- oder Kleinbuchstaben eintasten. In den Beispielen dieses Handbuchs werden meist Kleinbuchstaben verwendet.

Im EDIT-Modus werden die Leerzeichen Ihrer Eingaben generell ignoriert; Sie können soviele oder so wenige Leerzeichen verwenden wie Sie wünschen. Die ersten *zwei* Zeichen einer Zeile dürfen jedoch nicht durch ein Leerzeichen getrennt sein, und aufeinanderfolgende Ziffern einer Zahl dürfen nicht durch Leerzeichen getrennt sein.

Beispiel:

```
>sQ r (64)
```

Freie Wahl von Leerstellen.

```
8
```

Nach [RTN] wird die Wurzel aus 64 berechnet.

Zwischenräume und Kleinbuchstaben bleiben erhalten, wenn Sie Ihre Eingabe in Anführungszeichen einschließen. Es spielt keine Rolle, ob Sie einfache Anführungsstriche (' ', mit [SHIFT][7] erzeugt) oder doppelte Anführungsstriche (" ", mit [SHIFT][2] erzeugt) setzen, solange sie immer paarweise verwendet werden. In Anführungszeichen gesetzte Ausdrücke werden exakt in der eingegebenen Form interpretiert.*

Beispiel:

```
>'Text';"ubleibtu";'erhalten'
```

Die drei Strings in Anführungszeichen werden durch Semikolons getrennt.

```
Text bleibt erhalten
```

Beim Drücken von [RTN] werden die Strings wiedergegeben, wie sie eingegeben wurden.

In diesem Handbuch werden generell einfache Anführungszeichen benutzt, um Zeichenstrings zu kennzeichnen.

Eine Eingabehilfe ist die Wiederholungseingabe: Wenn Sie eine Taste oder Tastenkombination gedrückt halten, wird das entsprechende Zeichen nach einer kurzen Verzögerung wiederholt eingegeben. Zum Beispiel die Taste [B]:

```
>bbbbbbbbbbbbbbbbbbbbbbbbbbbbb
```

Wenn Sie [B] gedrückt halten, werden drei Fenster voller b's erzeugt. Die Zeile wird verschoben, sobald der Cursor am rechten Ende der Anzeige angelangt ist.

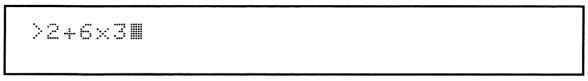
Systemtasten (wie [TIME]) und Editiertasten (wie [TAB]) wiederholen ihre Funktion ebenfalls, wenn sie gedrückt bleiben.

Die erste Anzeigeposition ist für das Aufforderungszeichen reserviert; die letzte, die 96ste Stelle ist für den Cursor reserviert. Damit haben Sie in jeder Zeile 94 Stellen zur Verfügung. Bei der Eingabe des 91sten Zeichens gibt der HP-75 ein Tonsignal – Sie können dann noch drei weitere Zeichen eingeben, bevor der Cursor das Zeilenende erreicht.

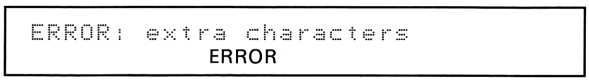
Probleme? ([SHIFT] [FET], [CLR], [ATTN], ERRN)

Falls der HP-75 während einer Eingabe ein Tonsignal ausgibt und die **ERROR** Anzeige erscheint, signalisiert er damit, daß er einen Befehl nicht interpretieren konnte oder eine nicht erlaubte Operation durchzuführen versuchte. In der Anzeige erscheint das Wort ERROR oder WARNING und eine kurze Meldung, und gewöhnlich wird auch Ihre letzte Eingabe zurückgegeben. Drücken Sie z.B. [CLR], geben Sie einen unrichtigen Ausdruck ein und drücken Sie [RTN]:

* Ausgenommen *Filenamen*, d.h. Namen in Anführungszeichen, die Files im Speicher spezifizieren. Benutzerfilenamen werden im Systemkatalog in Großschreibweise umgewandelt (Seite 45).



Der Multiplikationsoperator ist nicht x, sondern *.



Die Reaktion des HP-75. Die Fehlermeldung bleibt für etwa eine Sekunde in der Anzeige.

Danach wird der Originalausdruck angezeigt:

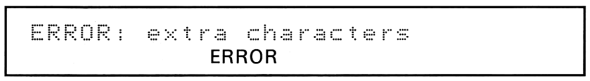
Der Cursor kehrt zur Stelle zurück, wo der HP-75 den ersten Fehler entdeckt hat.

Die Meldung `extra characters` bedeutet, daß der HP-75 die Zeichen nach `6`, dem letzten sinnvollen Zeichen in der Zeile, nicht interpretieren konnte.

Wenn ein Fehler auftritt, brauchen Sie sich nicht zu beunruhigen – Sie können das System mit keiner Tastenoperation beschädigen. Sie haben mehrere Möglichkeiten:

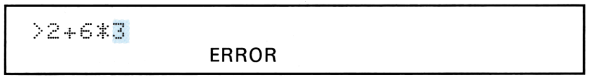
Nochmaliges Betrachten der Fehlermeldung. Halten Sie [SHIFT] gedrückt und drücken Sie zusätzlich die Taste [FET] (*fetch*) und *halten Sie die Taste* [FET] *gedrückt.*

Beispiel:

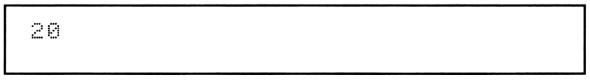


Mit [SHIFT] [FET] wird die derzeitige Fehlermeldung angezeigt.

Korrektur der Anzeige. Positionieren Sie den Cursor mit den Zeileneditiertasten (in diesem Fall die Taste [←], und geben Sie die notwendigen Korrekturen ein:



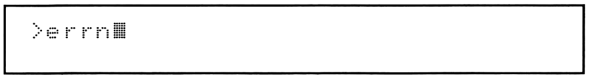
Die korrigierte Zeile. Drücken Sie [RTN] nach der Eingabe von *.



Der Ausdruck wird nun richtig als `2+(6*3)` aufgefaßt. (Die Multiplikation wird vor der Addition ausgeführt, wenn Sie die Reihenfolge nicht durch Klammern ändern).

Löschen der Anzeige Drücken Sie die Taste [CLR] oder [ATTN] und wiederholen Sie die Eingabe. Wenn die ERROR-Statusanzeige verschwindet, ist die Fehlerbedingung aufgehoben. Mit [SHIFT] [FET] erhalten Sie eine leere Anzeige.

Abruf der Fehlernummer. Jedem Fehler ist neben der Meldung eine Fehlernummer zugeordnet. Die Funktion `ERRN` gibt die Identifikationsnummer des zuletzt aufgetretenen Fehlers zurück:



Drücken von [RTN] ergibt die Anzeige 84, die dem Fehler `extra characters` zugeordnete Fehlernummer.

Anhang E enthält eine Tabelle der Fehlernummern, -meldungen und -bedingungen. Benutzen Sie die Funktion `ERRN`, um einen bestimmten Eintrag in der Tabelle zu finden.

Ein Fehler kann entweder eine Fehlermeldung `ERROR` oder eine Warnung (`WARNING`) verursachen. Eine Fehlerbedingung hält ein laufendes Programm an, während bei einer Warnung das Programm mit einem vorbestimmten Wert oder einem Ersatzwert weiter ausgeführt werden kann. In diesem Handbuch werden sowohl Fehler als auch Warnungen als *Fehler* bezeichnet, wenn nicht anders vermerkt.

Schreiben und Ausführen eines BASIC Programms (RTN, RUN, ATTN)

Die BASIC-Eingabeaufforderung im EDIT-Modus signalisiert, daß der Computer zur Eingabe von BASIC-Programmanweisungen bereit ist. Versuchen Sie dieses «Zufallsmusik»-Programm. Drücken Sie CLR und geben Sie ein:

```
>1 beep rnd*999,rnd*.5 @goto 1
```

Zwei Programmanweisungen sind in einer Zeile verbunden. Zwischenräume spielen keine Rolle.

```
>
```

Mit RTN wird die Programmzeile abgespeichert.

Schauen Sie sich das Programm mit ↑ noch einaml an.

```
>1 BEEP RND*999,RND*.5 @ GOTO 1
```

Der HP-75 hat das Programm in Großbuchstaben umgewandelt und die Zwischenräume zur besseren Lesbarkeit verändert.

Drücken Sie nun CLR oder ATTN, um die Anzeige zu löschen. Starten Sie das Zufallsmusikprogramm mit der Taste RUN oder mit der Eingabe `run`, gefolgt von RTN. Halten Sie das Programm mit ATTN an.

Beachten Sie, daß die Statusanzeige **PRGM** im Anzeigefenster andeutet, daß ein Programm abgearbeitet wird.

Während der Programmausführung ist das Tastenfeld für alle Tasten außer ATTN desaktiviert, um unbeabsichtigte Unterbrechungen zu vermeiden.

Geben Sie dem Programm nach dem Anhalten mit dem Befehl `RENAME` einen *Filenamen*. Tasten Sie ein:

```
>rename to 'singsong'
```

Mit RTN erhält das derzeitige Programm einen Namen, `SINGSONG`, der intern in Großbuchstaben umgewandelt wird.

Kopieren eines aufgezeichneten Programms (COPY CARD)

Eine der Funktionen des Kopierbefehls ist das Kopieren von auf Magnetkarten abgespeicherten Programmen in den Speicher. Ein Beispiel ist das Programm `FINANZ`, eines der drei Ihrem HP-75 beigelegten Fertigprogramme. Nehmen Sie die zwei `FINANZ` Magnetkarten aus dem mit Ihrem Gerät gelieferten Kartenhalter.

VORSICHT

Fassen Sie Magnetkarten nur an den Kanten an, um eine Verschmutzung der empfindlichen magnetischen Oberflächen und des Lesekopfes zu vermeiden. Die Sauberkeit der Karten ist für einwandfreies Lesen von großer Wichtigkeit.

Halten Sie die Karten unbedingt von starken Magnetfeldern fern. Die Karten würden vollständig nutzlos werden.

Schützen Sie Ihre Magnetkarten gegen Kratzer, Knicke und Schmutz, indem Sie sie sofort nach der Benutzung wieder in den Kartenhalter zurückstecken.

Die Anweisung COPY beginnt die Kartenleseroperation.

```
>copy card to 'finanz'
```

Der Befehl spezifiziert den Filenamen, den Sie dem Programm geben wollen. Drücken Sie [RTN].

```
Copy from card: Align & [RTN]
```

Die Reaktion zeigt, daß der HP-75 für eine Kartenleseroperation bereit ist.

Wenn Sie wollen, können Sie die Kartenleseroperation hier mit der Taste [ATTN] abbrechen. Andernfalls wartet der HP-75 auf das Einführen einer Karte und das Drücken von [RTN].

Jede Magnetkarte kann auf zwei *Spuren* beschrieben werden. Um die Information einer vollen Karte einzulesen, muß die Karte in beiden Richtungen gelesen werden. Die Reihenfolge der Karten und die Anzahl der Wiederholungen spielen keine Rolle.

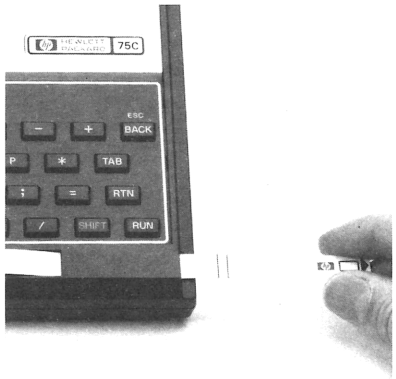
Führen Sie eine Karte mit der Schrift nach oben ein und schieben Sie sie in Richtung des Kartenleserpfeils. Schieben Sie die Karte soweit hinein, bis die rechte Markierung unter dem Eingabeschlitz verschwindet und die nächste Markierung sichtbar bleibt. Ungefähr 1 cm des Kartenendes sollte aus dem Ausgangsschlitz herausragen.

Wenn die Karte im Leser positioniert ist, drücken Sie [RTN]. Die Anzeige enthält:

```
Pull card...
```

Zum ersten Durchgang bereit.

Greifen Sie das rechte Kartenende mit Ihrem rechten Daumen und Zeigefinger und ziehen Sie langsam und gleichmäßig.



Sie sollten zum Durchziehen der Karte durch den Leser etwa solange brauchen wie zum Aufsagen des Namens «Hewlett-Packard». Die Kartengeschwindigkeit kann zwischen 13 und 76 cm/Sekunde liegen. Es kann nicht schaden, mit ein paar Versuchen ein Gefühl für die obere und untere Geschwindigkeitsgrenze zu entwickeln. Sie können die selbe Spur unbegrenzt oft durch den Kartenleser ziehen. Die Anzeige des HP-75 meldet sofort, ob der Lesedurchgang erfolgreich war oder nicht.

Beispiele:

```
WARNING: pulled too fast
```

Sie haben zu schnell gezogen.

```
Track 2 done; insert track 1
```

Spur 2 wurde eingelesen; die Karte kann zum Lesen der anderen Spur umgedreht werden.

```
WARNING: bad read/write
```

Die Karte wurde nicht richtig eingelesen. Wenn diese Warnung mehrmals auftrifft, sollten Sie die Karte reinigen (siehe Anhang B).

Nach der Anzeige der Meldung sehen Sie wieder:

```
Copy from card: Align & [RTN]
```

Sie können dieselbe Spur oder eine andere Spur des gleichen Programmes einlesen.

Drücken Sie vor jedem Durchgang [RTN]. Der HP-75 führt Sie dann mit Eingabeaufforderungen durch den Kartenleseprozess. Nach dem Einlesen aller vier Spuren des Programms erscheinen der Cursor und die Eingabeaufforderung wieder:

```
>
```

Diese Anzeige bedeutet, daß die Operation beendet wurde. Alle vier Spuren des Programms FINANZ wurden abgespeichert.

Abschnitt 3, «Editieren von Files», beschreibt den umgekehrten Prozeß, das Schreiben von Information vom Speicher auf Magnetkarten. In Abschnitt 8, «Kartenleser-Operationen», werden andere Möglichkeiten des Kartenlesers einschließlich Passwort und privater Magnetkarten besprochen.

Ausführen eines aufgezeichneten Programms (EDIT, [RUN], [ATTN])

Um das Programm FINANZ auszuführen, müssen Sie es erst mit Hilfe des Befehls EDIT (geben Sie edit ein, benutzen Sie nicht die [EDIT] Taste) im Programmspeicher suchen:

```
>edit 'finanz'
```

Mit [RTN] erhalten Sie den Katalogeintrag des Programms.

```
FINANZ     B 2248 13:52 07/13/82
```

Die Länge (in Bytes) und das Datum des Programmes können je nach Version variieren.

Hinweis: Falls bei der Eingabe des edit Befehls die Meldung ERROR: workfile name? auftritt, ist ein «Aufräumen» des Speichers erforderlich. Geben Sie purge [RTN] ein, um den momentanen Arbeitsspeicher zu löschen, und geben Sie danach den Befehl EDIT nochmals ein. (Siehe auch Abschnitt 3, Seite 63).

Sie haben mit dem FINANZ Programm zwei Möglichkeiten:

Spareinlagen. Das Programm berechnet aus einer Anfangsinvestition, der Anlagedauer und dem Zinssatz den *zukünftigen Wert* Ihrer Einlage.

Darlehen. Das Programm berechnet aus einem Darlehensbetrag, der Dauer des Darlehens und dem Zinssatz den Betrag Ihrer periodischen Zahlungen.

Beispiel: Wenn Sie ab heute monatlich 30 DM bei 7% monatlich abgerechneten Zinsen auf ein Konto, das derzeit 2000 DM enthält, einzahlen, wieviel wird in 9 Monaten auf diesem Konto sein? Geben Sie diese Informationen mit den Arithmetik-Tasten (Ziffern, Dezimalpunkt und Operatoren) ein.

Drücken Sie [RUN] oder geben Sie run [RTN] ein. Die Anzeige zeigt die **PRGM** Statusanzeige und den folgenden Inhalt:

```
~ F I N A N Z ~
PRGM
```

Sofort erscheint der Programmkopf gefolgt von den zwei Optionen.

Wollen Sie leihen oder sparen: PRGM	Geben Sie S oder s ein, um das Sparprogramm zu wählen, und drücken Sie dann [RTN].
********SPARGUTHABEN******** PRGM	Diese Meldung bestätigt Ihre Wahl.
Anfangswert: PRGM	Geben Sie 2000 [RTN] ein. Sie können Zahlen mit Dezimalpunkt und einen Operator (wie [+] oder [*]) eingeben, aber verwenden Sie keine Kommata.
Anzahl Perioden: PRGM	Geben Sie 9 ein, um neun Abrechnungsperioden zu spezifizieren.
Zinssatz/Periode: % PRGM	Wenn der Zinssatz jährlich 7% beträgt, dann ist der *monatliche* Zinssatz 7/12%. Geben Sie 7/12[RTN] ein.
Einzahlung/Periode: PRGM	Eingabe: 30[RTN]. (Es wird angenommen, daß die Einlage zu Beginn jeder Periode getätigt wird.)
Endwert: 2385.48 PRGM	Am Ende von 9 Monaten können Sie mit DM 2385.48 rechnen.

Die **PRGM** Statusanzeige bleibt gesetzt – das FINANZ Programm wartet auf weitere Anweisungen. Nach etwa fünf Sekunden zeigt die Anzeige:

(Fortsetzung mit bel. Taste) PRGM	Der HP-75 wartet auf eine Eingabe. Wird innerhalb einiger Sekunden keine Taste gedrückt, springt das Programm kontinuierlich zwischen diesen beiden Anzeigen hin und her.

Drücken Sie [ATTN], um das Programm anzuhalten. Wenn Sie eine weitere Spareinlage berechnen wollen, drücken Sie eine beliebige Taste außer [ATTN], z.B. [TAB]. Wenn keine Taste gedrückt wird, hält das Programm nach zwei Minuten automatisch an. Nach der Beendigung des Programms zeigt die Anzeige: Ende 'FINANZ' Programm.

********SPARGUTHABEN******** PRGM	Die Ausführung ist zum Beginn der Sparoption zurückgekehrt.

Dieses Mal hat der HP-75 die Zahlen des letzten Durchgangs gespeichert und zeigt sie an.

Beispiel:

Anfangswert: 2000.00 PRGM	Wenn der Betrag noch stimmt, übergeben Sie diese Eingabe, indem Sie nur [RTN] drücken. Andernfalls überschreiben Sie diesen Betrag und drücken Sie [RTN].
Anzahl Perioden: 9 PRGM	Sie können diese Zahl belassen oder überschreiben.

Alle Beträge werden mit zwei Dezimalstellen angezeigt. Der Zinssatz ist mit 1 bis 12 Stellen angezeigt. In diesem Beispiel:

Zinssatz/Periode: .583333333333	Eine Periode kann eine beliebige Zeitspanne sein. Das Beispiel benutzt monatliche Einlagen.

Halten Sie das Programm (mit [ATTN]) an, wenn Sie die bei einer Anleihe fälligen periodischen Zahlungen berechnen wollen, starten Sie es wieder (mit [RUN]), geben Sie ein `l` (*für leihen*) ein, und Sie sind im Geschäft.

Neudefinition von Tasten (DEF KEY)

Der Befehl `DEF KEY` ermöglicht Ihnen eine Neudefinition von Tasten als Eingabe- und Programmierhilfen.

Beispiel: Ändern Sie die nicht umgeschaltete Taste [Q], so daß Sie anstelle des kleinen `q` Ihren Namen anzeigt. Geben Sie die folgende Tastensequenz ein und drücken Sie [RTN]:

```
>def key 'q','Ihr Name';
```

Der Befehl `DEF KEY`. Beenden Sie den Befehl mit einem Semikolon, bevor Sie [RTN] drücken.

```
>
```

Die neue Belegung der Taste [Q] wird abgespeichert.

Wenn Sie jetzt [Q] drücken, erscheint Ihr Name. Wenn Sie die Taste gedrückt halten, wird Ihr Name über die ganze Anzeigezeile wiederholt. Sie können die ursprüngliche Tastenbelegung wieder erhalten, wenn Sie die Anzeige löschen und eingeben:

```
>purge keys
```

Mit [RTN] wird der Teil des Speichers, der Ihre Tastendefinition enthält, gelöscht.

Wenn Sie eine Taste umdefiniert haben, die Sie noch brauchen, geben Sie sie einfach als Großbuchstaben ein. Wenn Sie z.B. die Taste [P] umdefiniert haben, tasten Sie mit [SHIFT] [P] ein großes `P` ein.

Der Befehl `DEF KEY` ermöglicht 190 Tastendefinitionen. Sowohl *umgeschaltete Tasten* (z.B. [SHIFT][Q]) wie auch *Kontrolltasten* (z.B. [CTL][Q]) können vom Benutzer definiert werden. Siehe auch Abschnitt 10, «Neudefinition des Tastenfelds».

Eingabe eines Memo (EDIT, AUTO, PLIST)

Die Texteditierungsfähigkeiten des HP-75 erlauben Ihnen, im EDIT-Modus Korrespondenz aufzusetzen und zu korrigieren. Wenn Sie zum Beispiel ein kurzes Memo mit dem Namen `NOTCHEF` schreiben wollen, müssen Sie zuerst einen File mit diesem Namen im Speicher anlegen. Geben Sie die folgende Sequenz ein und drücken Sie [RTN]:

```
>edit 'notchef', text
```

Der Befehl `EDIT` legt den File `NOTCHEF` an und schaltet den HP-75 auf Editieren von Text.

```
NOTCHEF    T     0 10:16 02/05/83
```

Der Katalogeintrag mit dem Filenamen, des Filetyps (Text), der Filelänge (0 Bytes) und dem Zeitpunkt der Anlage.

Beim Drücken von [ATTN] oder einer beliebigen anderen Taste erscheint die Texteingabeaufforderung (:).

```
:
```

Mit [ATTN] erhalten Sie die Texteingabeaufforderung und den Cursor in der Anzeige.

Geben Sie nun `a.`, eine Abkürzung für `auto`, ein und drücken Sie [RTN].

```
:a.
```

Start der automatischen Zeilennumerierung.

```
:10
```

Die Numerierung beginnt bei Zeile 10.

Geben Sie dann das Memo zeilenweise ein und drücken Sie nach jeder Zeile [RTN].

```
:10 Von: Maier
```

Die erste Zeile des Memos. geben Sie Großbuchstaben mit der Taste [SHIFT] ein.

```
:20
```

Die Numerierung wird in Zehnerschritten inkrementiert.

```
:20 An: Chef
```

Die zweite Zeile.

```
:30
```

```
:30 Betreff: Computer
```

Dritte Zeile.

```
:40
```

```
:40 Vortreffliche Maschinen!
```

Letzte Zeile.

```
:50
```

Zeile 50 wird nicht mehr benutzt.

Beenden Sie die Eingabe mit [ATTN]. Geben Sie dann `delay 2` [RTN] ein, um eine angenehme Lesegeschwindigkeit zu erhalten. Führen Sie schließlich den Befehl `PLIST` (*print-list*) durch:

```
:plist
```

Drücken Sie [RTN], um die Eingabe zu überprüfen.

Der Befehl `PLIST` zeigt die Zeilen eines Files nacheinander im Abstand von zwei Sekunden ohne Zeilennummern an.

```
Von: Maier
An: Chef
Betreff: Computer
Vortreffliche Maschinen!
```

Letzte Zeile. Löschen Sie die Zeile mit [CLR].

Die Zeilen Ihres Memos werden genauso, wie Sie sie eingetastet haben, ohne Zeilennummer angezeigt.

Mit den Tasten [↓] und [↑] können Sie den Textfile in Einzelschritten zur Anzeige bringen. Sie können soviele Zeilen hinzufügen, wie Sie wollen. Geben Sie eine Zeilennummer und den Text ein, und drücken Sie [RTN]. Beachten Sie, daß eine Zeilennummer von der ersten Ziffer einer Zeile durch ein oder mehrere Leer- oder andere Zeichen getrennt sein sollte.

Beispiele:

```
:50 30 Tage im September.
```

Diese Zeile wird als Zeile 50 mit dem Inhalt `30 Tage im September` abgespeichert.

```
:5030 Tage im September.
```

Diese Zeile wird als Zeile 5030 mit nur dem Inhalt `Tage im September` abgespeichert.

Schließen Sie das Texteditierbeispiel mit einem `EDIT` Befehl ab, und kehren Sie zum `FINANZ` Programm zurück.

```
:edit 'finanz'
```

Abruf eines schon im Speicher stehenden Programms.

```
FINANZ     B 2384 13:52 07/13/83
```

Anzeige des Katalogeintrags des Programms.

Löschen Sie die Zeile mit [CLR]:

```
>
```

BASIC Eingabeaufforderung und Cursor erscheinen.

Mit dem Befehl EDIT können Sie jederzeit zu einem Programm oder einem Textfile zurückkehren.

Beispiel:

```
:edit 'notchef'
```

Geben Sie den Namen in Anführungszeichen in Groß- oder Kleinbuchstaben an.

```
NOTCHEF    T    90 10:16 02/05/83
```

Der Katalogeintrag zeigt, daß der File 90 Bytes Speicherplatz belegt.

Drücken Sie [CLR], und die Texteingabeaufforderung erscheint wieder. Geben Sie dann `edit 'finanz'` [RTN] ein, um wieder zum FINANZ Programm zurückzukehren.

Die Files SINGSONG, FINANZ und NOTCHEF bleiben im Speicher, bis Sie sie wieder löschen, zum Beispiel mit einem PURGE Befehl (Seite 50). Sie können soviele Files erzeugen und editieren, wie der Speicher aufnehmen kann. Der HP-75C kann über 150 Memos der Größe NOTCHEF aufnehmen. In Abschnitt 3, «Editieren von Files», finden Sie eine vollständige Anleitung zum Anlegen, Überprüfen und Bearbeiten von Files.

Kontrolle von HP-IL Einheiten

Im EDIT-Modus können Sie eine Vielzahl von HP-IL (Hewlett-Packard Interface Loop) Einheiten ansteuern, wie zum Beispiel das Video-Interface HP82163, den Thermodrucker HP82162A und das Digitalkassettenlaufwerk HP82161A. Der Anschluß dieser Peripheriegeräte wird im ersten Teil des Abschnitts 9, «HP-IL Operationen» beschrieben.

Die Vorwahltasten ([SHIFT], [CTL])

Der Computer kann trotz seines einfachen Tastenfeldaufbaus 256 Zeichen aller Art anzeigen. Die Leistungsfähigkeit des Tastenfelds wird durch verschiedenste Tastenkombinationen vervielfältigt. In allen Tastenfolgen werden die Taste [SHIFT] oder [CTL] oder beide verwendet.

So wie Vorsilben die Bedeutung von Worten verändern können, so verändern [SHIFT] und [CTL] die Funktion der Tasten, denen sie vorausgehen. Wie Vorsilben werden die Tasten [SHIFT] und [CTL] nicht allein benutzt; sie müssen gehalten werden, während eine zusätzliche Taste gedrückt wird. Die möglichen Tastenkombinationen sind in Anhang C zusammengestellt.

Das umgeschaltete Tastenfeld ([SHIFT] [LOCK])

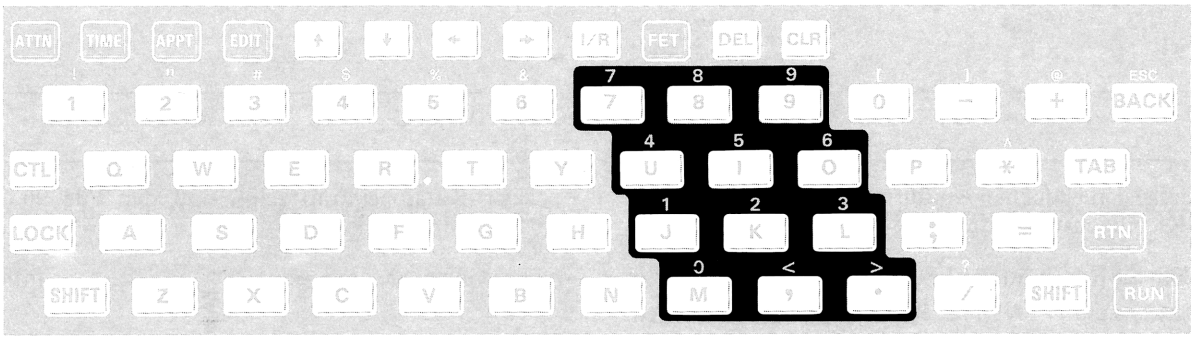
Wie bei einer Schreibmaschine werden die Tasten [A] bis [Z] in Groß- oder Kleinschreibung dargestellt, je nachdem, ob sie mit oder ohne der [SHIFT] Taste gedrückt wurden.

Sie können das Tastenfeld auf Großschreibung umschalten, indem Sie die Taste [SHIFT] gedrückt halten und zusätzlich die Taste [LOCK] drücken. Danach werden nicht umgeschaltete Buchstaben groß und *umgeschaltete* Buchstaben *klein* dargestellt. ([SHIFT] [LOCK] hat keine Auswirkungen auf Nicht-Buchstabentasten). Das Tastenfeld bleibt umgeschaltet, wenn der Rechner ausgeschaltet wird.

Um das Tastenfeld wieder zurückzuschalten, drücken Sie nur [LOCK]. Die Anweisung PUT, die in Abschnitt 13 behandelt wird, ermöglicht die Umschaltung des Tastenfelds in Programmen.

Das numerische Tastenfeld (CTL LOCK)

Um Ihnen bei der Zahleneingabe zu helfen, ist der HP-75 mit einem numerischen Tastenfeld ausgerüstet. In der folgenden Abbildung sehen Sie die Anordnung des Tastenfeldes. Sie können auch das Tastenfeldoverlay benutzen, das mit dem HP-75 geliefert wird.



Wenn Sie das numerische Tastenfeld benutzen wollen, halten Sie die Taste CTL gedrückt, während Sie LOCK drücken. Das numerische Tastenfeld, die Zifferntasten und die arithmetischen Operatoren sind dann aktiviert, nicht belegte Buchstabentasten sind desaktiviert und erzeugen einen Piepton, wenn sie dennoch gedrückt werden. Das numerische Tastenfeld beeinflußt die *umgeschalteten* Tasten nicht. Die nicht umgeschaltete Taste W erzeugt zum Beispiel einen Piepton, während das umgeschaltete W ein W anzeigt.

Anzeige von Sonderzeichen (CTL)

Wenn Sie die Taste CTL gedrückt halten, während Sie eine Buchstabentaste (wie D) oder einige andere Tasten (wie das Leerzeichen) drücken, wird dadurch ein Sonderzeichen (*Kontrollzeichen*) erzeugt. Halten Sie CTL und drücken Sie zusätzlich C und dann Q: die Zeichen ◻ und Ω erscheinen. Der vollständige Zeichensatz des HP-75 wird in Abschnitt 2 besprochen.

Allgemeine Information

Sperren des HP-75 (LOCK)

Mit dem Befehl LOCK können Sie den HP-75 gegen ungewünschte Benutzung durch Dritte, die Ihr Paßwort nicht kennen, schützen. Das Paßwort kann eine beliebige Kombination von bis zu 90 Zeichen sein, wenn auch der Computer nur die ersten 8 Zeichen prüft. Wir empfehlen Ihnen, ein leicht zu merkendes Paßwort zu verwenden, da die Sperre *absolut* ist. Wenn Sie das Paßwort vergessen, müssen Sie den HP-75 zurücksetzen.

Schließen Sie bei der Eingabe des LOCK-Befehls das Paßwort in Anführungszeichen ein.

Beispiel:

```
>lock 'Ben'
```

Paßwort in Anführungszeichen.

```
>
```

Gesichert mit dem Paßwort BEN.

Der HP-75 fragt bei jedem Einschalten nach dem derzeitigen Paßwort. Schalten Sie das Gerät mit [SHIFT] [ATTN] aus und mit [ATTN] wieder ein, um dies zu demonstrieren.

```
password? 
```

Sofort nach dem Einschalten fragt der HP-75 Ihr Paßwort ab.

Geben Sie ein falsches Paßwort ein und drücken Sie [RTN], um zu sehen, was passiert:

```
password? ben
```

```

```

B muß ein Großbuchstaben sein.

Das Gerät schaltet sich selbst wieder aus.

Um den HP-75 wieder bedienen zu können, müssen Sie das Paßwort *exakt* wie beim LOCK Befehl eingeben.

Beispiel:

```
password? Ben
```

```
>
```

Korrekte Eingabe. Anführungszeichen werden nicht benötigt.

Eingabeaufforderung und Cursor erscheinen.

Der HP-75 schaltet sich nur dann eingabebereit in den EDIT-Modus, wenn die ersten acht (oder weniger bei kürzerem Paßwort) Eingabezeichen mit den ersten acht Zeichen des Paßwortes übereinstimmen. Geben Sie LOCK '' [RTN] *ohne Leerzeichen* ein, wenn Sie ein eingegebenes Paßwort entfernen wollen. Dies sichert den HP-75 mit einem nullstelligen, nichtexistenten Paßwort.

Der HP-75 kann auch unter Programmkontrolle mit der Anweisung LOCK gesichert werden (Seite 174).

Desaktivieren der Ausschaltautomatik (STANDBY ON, STANDBY OFF, BYE)

Der Befehl STANDBY ON läßt Sie die normale fünfminütige Ausschaltfrist überschreiten.

```
STANDBY ON
```

Wenn Sie standby on [RTN] eingeben, bleibt der HP-75 unbegrenzt lange eingeschaltet. Um die fünfminütige Ausschaltfrist wieder herzustellen, müssen Sie standby off [RTN] eingeben.

```
STANDBY OFF
```

Der Vorteil von STANDBY ON liegt darin, daß die Anzeige eingeschaltet bleibt, auch wenn Sie den Computer nicht benutzen. So kann der HP-75 beispielsweise als Tischuhr dienen. Der Vorteil des standby off Zustands ist, daß die Batterien geschont werden, wenn der Computer nicht ans Netz angeschlossen ist. Voll geladene Nickel-Cadmiumbatterien erlauben ungefähr 20 Stunden STANDBY ON Betrieb.

Wenn Sie den Rechner ausschalten (mit [SHIFT] [ATTN] oder der Eingabe BYE[RTN]) bleibt der vorhergehende STANDBY Zustand beim Wiedereinschalten mit [ATTN] erhalten.

```
BYE
```

Der Befehl BYE kann zusammen mit Programmtimern zum regelmäßigen Ein- und Ausschalten des HP-75 unter Programmkontrolle benutzt werden (Seite 187).

Beachten Sie, daß die STANDBY OFF Ausschaltfrist auf 10 Minuten verlängert wird, wenn Sie ein Ergebnis in der Anzeige stehen haben. Wenn Sie zum Beispiel 60*60*24 RTN eingeben und die 86400 in der Anzeige stehen lassen, bleibt die Anzeige für fünf Minuten erhalten; danach erscheinen der Cursor und die Eingabeaufforderung; und nach weiteren fünf Minuten schaltet sich der Computer aus. Bei STANDBY ON bleibt ein Ergebnis unbeschränkt lange in der Anzeige stehen.

Der Tonsignalgeber (BEEP, BEEP OFF, BEEP ON)

Die Anweisung BEEP erzeugt über den eingebauten Lautsprecher einen hörbaren Ton. Sie haben drei Formen dieser Anweisung zur Verfügung:

```
BEEP
BEEP Frequenz in Hertz
BEEP Frequenz in Hertz, Tondauer in Sekunden
```

Die Eingabe von beep RTN erzeugt einen 0.1 Sekunden dauernden 1400 Hz-Piepton. Wenn Sie nur die Frequenz definieren, ertönt das Signal für ungefähr 0.1 Sekunden.

Beispiel:

```
>beep 261
```

Nach RTN ertönt ein mittleres C für ungefähr 0,1 Sekunden.

Sie können eine beliebige Frequenz eingeben; Sie erhalten jedoch die beste Tonqualität zwischen 100 und 1400 Hertz.

Der Zeitparameter setzt die Tondauer auf die Zehntelsekunde genau.

Beispiel:

```
>beep 220, 1.5
```

Wenn Sie RTN drücken, ertönt für 1,5 Sekunden das A unter dem mittleren C.

Frequenzen unter 10 Hz und negative Zeitwerte erzeugen ein Knacken des Lautsprechers. Frequenzen und Zeiten können auch über numerische Ausdrücke spezifiziert werden (zum Beispiel BEEP 440*2^(1/12),L). Die obere Zeitgrenze beträgt 2^{26} Sekunden. Werte über 2^{26} Sekunden werden durch diese Grenze ersetzt.

Mit ATTN können Sie jedes Tonsignal unterbrechen.

Hinweis: Der Lautsprecher hat, besonders bei niedrigen Frequenzen, einen beträchtlichen Stromverbrauch.

Mit zwei Befehlen können Sie das Tonsignal aktivieren und desaktivieren.

```
BEEP OFF
```

```
BEEP ON
```

Geben Sie `beep off` RTN ein, um das Tonsignal auszuschalten. Danach werden fällige Termine*, Zeilenenden und Fehlermeldungen nicht mehr angekündigt und `BEEP` Anweisungen nicht mehr ausgeführt. Nach `BEEP OFF` bleibt das Tonsignal solange desaktiviert, bis Sie den Befehl `BEEP ON` eingeben.

Abkürzung von Schlüsselworten (·)

Viele der Worte, die der HP-75 erkennt (sog. *Schlüsselworte*), können zur Zeitersparnis abgekürzt werden. Um einen Ton zu erzeugen, können Sie zum Beispiel `be.` anstelle von `beep` eingeben. Hier sind die kürzestmöglichen Abkürzungen der bisher verwendeten Schlüsselworte:

Schlüsselwort	Kürzestmögliche Abkürzung
AUTO	a.
BEEP	be.
COPY	co.
EDIT BASIC	e.ba.
EDIT TEXT	e.t.
PLIST	pl.
PURGE	pu.
RENAME	ren.
RUN	r.
STANDBY OFF	s.off
STANDBY ON	s.on

Beachten Sie, daß alle Abkürzungen mit Punkten enden, und daß Sie bei der Abkürzung eine längere Kurzform als die angegebene verwenden können. Zum Beispiel ist `pli.` eine erlaubte Abkürzung für `PLIST`. Die folgenden Beschränkungen gelten bei Abkürzungen:

- Sie müssen *mindestens* so viele Zeichen wie in der kürzesten unterscheidbaren Abkürzung verwenden. Zum Beispiel ist `p.` keine gültige Abkürzung für `PLIST`.
- Der Punkt darf nicht den letzten Buchstaben des Schlüsselwortes ersetzen. Zum Beispiel ist `aut.` keine gültige Abkürzung für `AUTO`.
- Ein abgekürztes Schlüsselwort darf keine Leerzeichen enthalten.

Ein Schlüsselwort kann in jeder Anzeigezeile durch eine Abkürzung ersetzt werden; in diesem Handbuch werden jedoch alle Schlüsselworte zur besseren Lesbarkeit ausgeschrieben. Im Anhang D finden Sie eine vollständige Liste der Abkürzungen der Schlüsselworte des HP-75.

Ausblick

Sie haben in diesem Abschnitt schon einiges über den HP-75 gelernt. Weitere Informationen folgen, die Sie aber nicht vollständig zu lesen brauchen. Wir empfehlen Ihnen, diesen Abschnitt noch einmal zu lesen und die Beispiele der Abschnitte 2 und 3 zu bearbeiten, aber von da an haben Sie die Wahl. Sie können, wenn Sie wollen, nach Abschnitt 3 direkt bei der Programmierung, Teil III, weiterarbeiten.

Wenn Sie sich über die exakte Bedeutung der hier benutzten Terminologie nicht sicher sind, sollten Sie in Anhang G nachschlagen. Anhang G enthält ein Glossar, das viele der in diesem Handbuch verwendeten Ausdrücke definiert.

---

* Sie haben 10 verschiedene Alarme (0 bis 9) zur Verfügung, um Termine anzukünden. Alarm Nummer 6 erzeugt immer ein Tonsignal, auch nach dem Befehl `BEEP OFF`.

Syntax-Richtlinien

Syntax ist die Art, in der Anweisungen eingegeben werden müssen, damit der Computer ihre Bedeutung versteht. Die folgenden Syntaxregeln werden in diesem Handbuch durchgehend eingehalten.

PUNKTMATRIX	In Punktmatrix (wie PLIST) gesetzter Text ist in der angegebenen Weise in Groß- oder Kleinbuchstaben einzugeben. In den Beispielen werden Befehle, Anweisungen und Funktionen in Kleinbuchstaben eingegeben und intern in GROSSBUCHSTABEN umgewandelt.
Kursiv	Die Parameter einer Anweisung, wie die *Frequenz in Hertz* bei der Anweisung BEEP, sind kursiv gesetzt.
' ',""	Filenamen und andere Zeichenstrings können in einfache oder doppelte Anführungszeichen eingeschlossen und klein oder groß geschrieben werden. In den Beispielen werden einfache Anführungszeichen verwendet. Filenamen in Anführungszeichen werden intern in Großbuchstaben umgewandelt.
[]	Eckige Klammern schließen optionale Ausdrücke ein.
...	Drei Punkte deuten an, daß die optionalen Teile in einer Klammer wiederholt werden können.
übereinander gesetzt	Wenn zwei oder mehrere Ausdrücke übereinander gesetzt sind, ist einer (und nur einer) davon zu spezifizieren.
oder	Sind zwei oder mehrere Ausdrücke durch «oder» getrennt, können beide mehrmals verwendet werden.

Im *HP-75 Referenzhandbuch* finden Sie eine Aufstellung der Syntax aller HP-75 Anweisungen. Zusätzlich enthält auch Anhang H, «Syntax-Zusammenfassung», eine detaillierte Beschreibung der Anweisungssyntax des HP-75 und eine vollständige Zusammenstellung der Syntax-Flußdiagramme. Anhang H ist ein nützliches und präzises Nachschlagewerk für alle Befehle und Anweisungen Ihres Computers. Vielleicht möchten Sie diesen Anhang lesen, um sich mit ihm vertraut zu machen, bevor Sie hier im Handbuch weiterarbeiten. Schlagen Sie beim Durcharbeiten des Handbuchs immer in Anhang H nach, wenn Sie Fragen zur Syntax des HP-75 haben sollten. Anhang H ist auch als handliche Referenz beim Schreiben von Programmen sehr nützlich.

Abschnitt 2

Tastenfeld- und Anzeigesteuerung

Inhalt

Einführung

In diesem Abschnitt werden die Editiertasten des HP-75 und die Tastenfolgen zur Kontrolle der Anzeigezeile diskutiert. Weiterhin wird der vollständige Zeichensatz mit 256 Zeichen vorgestellt. Sie sollten wissen, wie Zeichen hinzugefügt, gelöscht und eingefügt werden, um den Computer wirkungsvoll nutzen zu können.

Das Tastenfeld

Das Tastenfeld besitzt 65 Tasten, die drei Gruppen zugeordnet werden können:

- Schreibmaschinentasten. Diese bestehen aus 45 Buchstaben- ([A]), Ziffern- ([9]) und Symboltasten ([*]) und der Leertaste.
- Systemtasten. Diese sieben Tasten sind [ATTN], [RTN], [TIME], [APPT], [EDIT] [RUN] und [FET]. Sie haben diese Tasten schon in Abschnitt 1 benutzt und werden sie auch weiterhin zur Steuerung des HP-75 Betriebssystems anwenden.
- Editiertasten. Die 13 Editiertasten und eine große Zahl von Tastenkombinationen geben Ihnen eine Vielzahl von Kontrollmöglichkeiten bei der Eingabe von Information in die Anzeige. Beispiele aus Abschnitt 1 sind [BACK], die den Cursor zurücksetzt, und [SHIFT] [LOCK], die das Tastenfeld auf Großschreibung umschaltet.

Die Abbildung des Tastenfelds auf Seite 6 zeigt die Anordnung der Schreibmaschinen-, System- und Editiertasten.

Die Editiertasten

Die Editiertasten arbeiten im allgemeinen in allen drei Modi TIME, APPT und EDIT* auf gleiche Weise. In der Kombination mit den Tasten [SHIFT] und [CTL] können die meisten Editiertasten noch eine zweite und eine dritte Funktion ausführen. Wie die Schreibmaschinentasten wiederholen auch die Editiertasten ihre Funktion, wenn sie gedrückt bleiben.

Da in diesem Abschnitt Programmanwendungen behandelt werden, steht in diesem Abschnitt die BASIC Eingabeanforderung (>) in der Anzeige, obwohl die Texteingabeanforderung (⋮) den gleichen Zweck erfüllen würde. Der HP-75 kann Hunderte von Programmzeilen speichern, aber die Besprechung ist auf eine einzige Anzeigezeile konzentriert, die momentane Zeile, die alle Anzeigezeilen im EDIT-Modus repräsentiert.

Bewegen des Cursors in der Anzeige ([SHIFT] [←], [SHIFT] [→], [CTL] [←], [CTL] [→])

Der Linkspfeil ([←]) und der Rechtspfeil ([→]) bewegen den Cursor in der Anzeigezeile, ohne den Inhalt der Anzeige zu verändern. Sie können den Cursor auf vier Weisen mit der Taste [←] nach links bewegen:

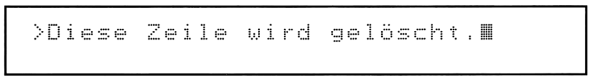
- Drücken Sie wiederholt [←].
- Drücken Sie [←] und halten Sie die Taste für eine fortlaufende Bewegung gedrückt.
- Drücken Sie [CTL] [←], um den Cursor um 32 Zeichen (ein Anzeigefenster) nach links zu bewegen. Ist der Cursor weniger als 32 Zeichen vom Anfang der Zeile entfernt, wird er auf den Zeilenanfang gesetzt.
- Setzen Sie den Cursor mit [SHIFT] [←] an den Anfang der Zeile.

Mit den Tastenkombinationen [→], [CTL] [→] und [SHIFT] [→] bewegen Sie den Cursor entsprechend nach rechts.
Beachten Sie, daß [TAB] und [SHIFT] [TAB] im TIME- und APPT-Modus zur Vorwärts- und Rückwärtsbewegung des Cursors benutzt werden. Im EDIT-Modus ist [TAB] ohne Wirkung, und [SHIFT] [TAB] hat die gleiche Wirkung wie [SHIFT] [←].

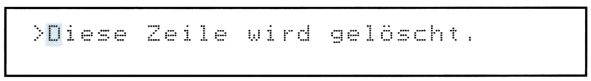
Löschtasten ([BACK], [SHIFT] [BACK], [DEL], [SHIFT] [DEL])

Bei gedrückter Taste [BACK] wird der Cursor nach links bewegt und löscht dabei ein Zeichen nach dem anderen, bis er den Zeilenanfang erreicht. [SHIFT] [BACK] hat die gleiche Wirkung.

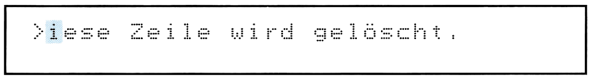
Mit der Taste [DEL] können Sie ein Zeichen aus der Anzeige löschen, ohne dabei ein Leerzeichen zu erzeugen. Wenn Sie die Taste gedrückt halten, wird der Löschvorgang wiederholt. Geben Sie zum Beispiel ein:



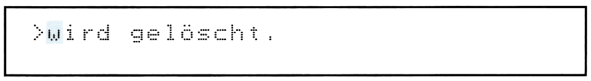
Drücken Sie nun [SHIFT] [←], um den Cursor an den Zeilenanfang zu setzen.



Beim ersten Drücken von [DEL] wird das D gelöscht.



Halten Sie die Taste nun für kontinuierliches Löschen gedrückt.

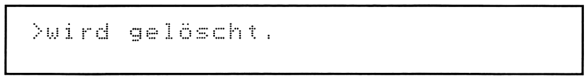


Die Taste [DEL] löscht ein Zeichen nach dem anderen, wobei der Zeileninhalt nach links geschoben wird.

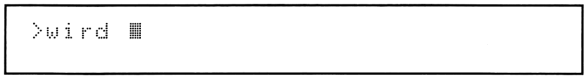
* Es gibt sinnvolle Ausnahmen. Beispiel: Die Taste [CLR] löscht im EDIT-Modus eine Anzeigezeile und erzeugt im APPT-Modus eine unbeschriebene APPT-Maske.

[SHIFT] [DEL] beschleunigt den Löschvorgang. Diese Tastenkombination löscht alle Zeichen rechts der momentanen Cursorposition (einschließlich des Zeichens, auf dem der Cursor momentan positioniert ist).

Beispiel:



Bewegen Sie den Cursor mit [→] nach rechts.



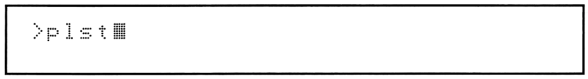
Mit [SHIFT] [DEL] wird bis zum Zeilenende gelöscht.

Die Einfügungs/Ersetzungstaste ([I/R])

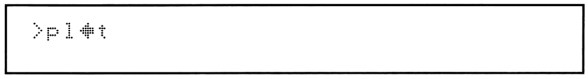
Der HP-75 verfügt über zwei Cursor, um Ihnen die Position des nächsten eingegebenen Zeichens anzuzeigen. Mit der Einfügungs/Ersetztaste [I/R] können Sie zwischen beiden hin- und herschalten:

- Der normale Cursor ist der Ersetzungscursor, das blinkende Zeichen ▒.
- Der Einsetzungscursor, ein blinkendes ◆, erscheint nach einem Drücken von [I/R] und verschwindet nach einem weiteren Drücken dieser Taste wieder.

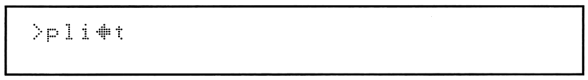
Der Einfügungscursor ermöglicht Ihnen das Einfügen von Zeichen innerhalb einer schon geschriebenen Zeile. Löschen Sie die Anzeige (mit [CLR]) und geben Sie einen fehlerhaften Befehl ein:



Es fehlt ein i im Befehl PLIST. Drücken Sie [I/R] und positionieren Sie den Einfügungscursor mit [←].



Das ◆ zeigt auf die Stelle, wo das nächste Eingabezeichen erscheinen wird. Geben Sie nun das i ein.

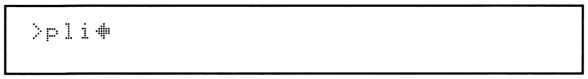


Das vor dem ≤ eingesetzte i.

Drücken Sie nochmals [I/R], um wieder den Ersetzungscursor anzuzeigen. Oder drücken Sie stattdessen [RTN], um PLIST mit dem Einfügungscursor in der Anzeige auszuführen.

Der Einfügungscursor kann zur Editierung des Notizfeldes im APPT-Modus benutzt werden. Im TIME-Modus hat [I/R] keine Wirkung.

Sie können mit jeder Editiertaste oder -Tastenfolge auch den Einfügungscursor steuern. Drücken Sie zum Beispiel [SHIFT] [DEL]:



Löscht alle Zeichen unter und nach dem Cursor.

Mit [BACK] löschen Sie das Zeichen links des Einfügungscursors – das Zeichen, auf das ◆ zeigt – wobei das Zeilenende nach links geschoben wird.

Anzeigen von Informationen

Anzeigewiederholungen ([RTN], [→], [←])

Ziffern und Zeichen können vom HP-75 bei der Eingabe reproduziert werden. Wenn Sie im EDIT-Modus eine Zahl und dann [RTN] eingeben, versucht der HP-75 diese Zahl zuerst als Zeilennummer zu interpretieren (eine Zahl zwischen 1 und 9999). Ist die Zahl größer als 9999, oder weist sie einen Dezimalpunkt oder ein führendes Vorzeichen auf, dann wird diese Zahl zur Kontrolle noch einmal angezeigt.

Beispiele:

```
>123456.7
```

Eingabe einer Dezimalzahl.

```
123456.7
```

Anzeige der Zahl.

```
>-3045
```

Eingabe einer negativen Zahl.

```
-3045
```

```
>12
```

Eingabe einer ganzen Zahl zwischen 0 und 9999.

```
>
```

Diese Eingabe wird als BASIC-Zeile (wenn auch leer) aufgefaßt und abgespeichert.

Über das Tastenfeld eingegebene Ausdrücke werden berechnet, und das Ergebnis wird angezeigt.

Beispiel:

```
>-6-12
```

```
-18
```

Ein einfacher Ausdruck.

Wenn Sie Ausdrücke mit Komma oder Semikolon voneinander trennen, kann der HP-75 auch mehrere Ausdrücke gleichzeitig berechnen. Die Ergebnisse von durch Kommata getrennten Ausdrücken erscheinen auseinander gesetzt, während die Ergebnisse durch ein Semikolon getrennter Ausdrücke nebeneinander angezeigt werden.

Beispiele:

```
>1.17,5.14
```

```
1.17                    5.14
```

Trennen der Ausdrücke durch ein Komma.

```
>1.17;5.14;2+1
```

```
1.17  5.14  3
```

Trennen der Ausdrücke durch ein Semikolon.

Auf dem HP-75 werden Zahlen mit einer Leerstelle für das Vorzeichen (leer für positiv, minus für negativ) und mit einem nachlaufenden Leerzeichen dargestellt. Beachten Sie, daß nur die letzten Ergebnisse angezeigt werden, wenn die Länge des Anzeigefensters nicht für alle Ergebnisse ausreicht.

Auch in Anführungszeichen (' ' oder " ") eingeschlossene Zeichen werden vom Rechner protokolliert. Mit RTN werden die Zeichen des Strings angezeigt.

Beispiel:

```
>'Wettervorhersage: sonnig.'
```

```
Wettervorhersage: sonnig.
```

In Anführungszeichen eingeschlossener String.

Protokollierung ohne Anführungszeichen.

```
>'Sagte sie "Ja"?'
```

Zwei Paar Anführungszeichen. Beachten Sie, daß das innere Paar vom äußeren verschieden sein muß.

```
Sagte sie "Ja"?
```

Das innere Paar bleibt erhalten.

Zahlen und in Anführungszeichen gesetzte Strings können zu gemischten Ausdrücken kombiniert werden.

Beispiel:

```
'Diese Woche nur';4;"Werktage."
```

Zwei Strings in Anführungszeichen und eine Zahl, durch Semikolons getrennt.

```
Diese Woche nur 4 Werktage.
```

Beachten Sie, daß die Tasten [←] und [→] allein oder in Kombination mit den Umschalttasten [SHIFT] und [CTL] zum Verschieben der in der Anzeige stehenden Zeichen benutzt werden können. Drücken Sie zum Beispiel dreimal [←]:

```
se Woche nur 4 Werktage.
```

Mit [←] und [→] wird das angezeigte Ergebnis hin- und hergeschoben.

Wenn Sie eine andere Taste drücken, erscheint wieder die Eingabeaufforderung und der Cursor.

Über das Tastenfeld berechnete Ergebnisse werden automatisch protokolliert; über Programm erzeugte Resultate können mittels der Anweisung DISP oder PRINT ausgegeben werden. Schlagen Sie dazu in Abschnitt 11, «Anzeigen und Drucken von Information», Seite 166 nach.

Abruf der letzten Eingabe ([CTL] [FET])

Bei der Eingabe wird jedes in der Anzeigezeile erscheinende Zeichen zusätzlich in einen temporären *Eingabebuffer*, einem 96-Zeichen- Speicher zur Speicherung des Anzeigeinhalts, abgelegt. Mit der Taste [RTN] beenden Sie die Eingabe in den Eingabebuffer. Im EDIT-Modus können Sie mit gehaltenem [CTL] und zusätzlich gedrückter Taste [FET] den derzeitigen Inhalt des Eingabebuffers in die Anzeige holen. Wenn Sie jetzt zum Beispiel [CTL] [FET] drücken, sehen Sie:

```
'Diese Woche nur';4;"Werktage."
```

Die letzte Zeile vor [RTN].

Die Zeile wird linksbündig angezeigt; der Cursor steht ganz links. Die Zeile kann editiert und ausgeführt werden, auch wenn die Eingabeaufforderung nicht sichtbar ist.

Diese Tastenkombination kann Ihnen Eingabezeit sparen, da Sie lange Tastenfeldeingaben einfach abrufen können und nicht nochmals einzugeben brauchen.

Nach dem Drücken von [RTN] werden die alten Zeichen im Eingabebuffer durch die neu eingegebenen ersetzt. Mit [CTL] [FET] können Sie den veränderten Bufferinhalt zur Anzeige bringen. Drücken Sie jetzt [CLR] und geben Sie ein:

```
'Jede
```

Diese neuen Zeichen werden in den Eingabebuffer geschrieben. Prüfen Sie mit [CTL] [FET] nach.

```
'Jedee Woche nur';4;"Werktage."
```

Die abgerufene Zeile enthält die zuletzt eingegebenen Zeichen.

Hinweis: Neben [RTN] beenden auch [ATTN], [↑], [↓] und [RUN] die Zeicheneingabe in den Eingabebuffer.

Die folgenden Befehle greifen auf den Eingabebuffer zu; Ihre letzte Eingabe geht dabei verloren: FETCH, FETCH KEY, LIST, LOCK, PLIST und TRANSFORM.

Die Verzögerungsrate (DELAY)

Der Befehl DELAY bestimmt die Anzeigegeschwindigkeit von Fehlermeldungen, Kartenlesermeldungen, HP-IL Meldungen und Programmausgaben auf die Anzeige und andere Sichtgeräte.

DELAY *Anzahl Sekunden*

Die *Anzahl Sekunden* ist ein numerischer Ausdruck, der auf die Zehntelsekunde genau die Anzeigezeit einer Zeile angibt, bevor die nächste Zeile eingelesen wird. Im Einschaltzustand ist DELAY auf 1 Sekunde voreingestellt; im Programm FINANZ wird die Verzögerung auf 0 Sekunden geändert.

Der Parameter kann Werte zwischen 0 und 2^{26} Sekunden annehmen. Werte außerhalb dieser Grenzen werden automatisch durch den entsprechenden Grenzwert ersetzt.

Beispiel:

```
>delay 2
```

Diese Eingabe, gefolgt von RTN, setzt die Verzögerungsrate auf 2 Sekunden.

Beachten Sie, daß DELAY auf die Anzeige einzelner Zeilen keine Auswirkung hat. Protokollierte Zeilen (Seite 36) zum Beispiel bleiben bis zum nächsten Tastendruck oder bis zum Ausschalten des HP-75 angezeigt.

Sie können die Verzögerung durch Drücken einer beliebigen Taste umgehen. Bei jedem Tastendruck wird die derzeitige Anzeigezeile sofort durch die nächste Zeile ersetzt. Wenn Sie zum Beispiel beim Listen des Files NOTCHEF die Taste TAB gedrückt halten, werden die Zeilen ohne Verzögerung angezeigt.

Die momentane Verzögerungsrate bleibt solange wirksam, bis ein neuer DELAY Befehl ausgeführt wird.

Zeilenlänge (WIDTH, PWIDTH)

Mit den Befehlen WIDTH und PWIDTH können Sie die in einer Zeile anzuzeigende oder zu druckende Anzahl Zeichen bestimmen.

WIDTH *Anzahl Zeichen*

PWIDTH *Anzahl Zeichen*

Der Befehl WIDTH spezifiziert die Zeilenlänge von Anzeigeausgaben; der Befehl PWIDTH setzt die Zeilenlänge von Druckerausgaben. Im Einschaltzustand sind WIDTH und PWIDTH auf 32 Zeichen, die Länge des Anzeigefensters, gesetzt. Diese Befehle beeinflussen das Verhalten des Anzeigefensters bei der *Eingabe* nicht.

Als WIDTH oder PWIDTH Parameter können beliebige Ausdrücke verwendet werden. Die jeweils geeignete Länge hängt von der Ausgabeeinheit ab. Für den Thermodrucker HP82162A ist 24 oder ein Vielfaches davon ein sinnvoller Wert, da der Drucker Druckzeilen von 24 Zeichen erzeugt.

Der HP-75 sendet soviele Zeichen an die Anzeige oder den Drucker, wie die momentane Einstellung von WIDTH oder PWIDTH bestimmt, und sendet danach Wagenrücklauf/Zeilenvorschub-Codes, die den Anfang einer neuen Zeile signalisieren.

Beispiel: Zeigen Sie das Datum und die Uhrzeit im EDIT-Modus an, so daß die Information in Gruppen von je drei Zeichen nacheinander in der Anzeige erscheinen. Setzen Sie die Zeilenlänge der Anzeige mit `width 3` RTN. Benutzen Sie dann die Funktionen `DATE$` und `TIME$` (Seite 98). Geben Sie ein:

```
>date$;time$
```

Nach RTN werden das Datum und die Zeit entsprechend den derzeitigen Werten von `WIDTH` und `DELAY` angezeigt.

Das Beispiel zeigt, daß bei Werten kleiner als 32 angezeigte Zeilen in kleinere Teile zerlegt werden. Bei Werten über 32 werden lange Zeilen über die Anzeige geschoben. Wenn Sie Werte über 96 spezifizieren, überschreiben sich die nachfolgenden Zeichen in der letzten Stelle.

Die `WIDTH` und `PWIDTH` Parameter werden zu ganzzahligen Werten gerundet; Werte von `0` oder `1` spezifizieren eine einspaltige Anzeige. (Negative Werte und Werte größer als 255 werden durch 255 ersetzt, sind aber begrenzt durch die Zeilenlänge der Anzeige und des Druckers). Mit `WIDTH INF` und `PWIDTH INF` werden unbegrenzt viele Zeichen in eine Zeile gesetzt – die Ausgabe wird nie auf die nächste Zeile gelenkt.

`WIDTH` oder `PWIDTH` Vorgaben bleiben so lange erhalten, bis Sie neue Werte spezifizieren. Sie sollten den HP-75 jetzt auf seine ursprüngliche Einstellung zurücksetzen, indem Sie `width 32` RTN eingeben. `DELAY` und `WIDTH` Befehle sind in Programmen nützlich, um Anfangszustände der Anzeige zu spezifizieren.

Setzen des rechten Rands (`MARGIN`)

Nach einem Zurücksetzen erzeugt der HP-75 bei jeder Eingabe des 91. Zeichens einer Zeile ein Tonsignal. Mit dem Befehl `MARGIN` können Sie eine neue Position angeben, bei der ein Tonsignal das Zeilenende ankündigen soll.

`MARGIN` *Anzahl Zeichen*

Beispiele:

```
>margin 26
```

Das Tonsignal erklingt bei der Eingabe des 26. Zeichens.

```
>margin 75
```

Das Tonsignal erklingt bei der Eingabe des 75. Zeichens.

Beachten Sie, daß das Tonsignal eine Erinnerung an das Zeilenende, und nicht eine Begrenzung der Zeile darstellt. Das Signal hängt in Wirklichkeit von der Stellung des Cursors ab und ist im Falle des Einfügungscursors unabhängig von der Zeilenlänge. Sie können unabhängig von der `MARGIN` Einstellung noch immer 94 Zeichen (96 Spalten minus eine Stelle für die Eingabeaufforderung und eine Stelle für den Cursor) in eine gegebene Zeile eintasten.

Die Anzahl der Zeichen kann durch einen beliebigen numerischen Ausdruck (wie z.B. `90-15`) spezifiziert werden; der Ausdruck wird auf einen ganzzahligen Wert gerundet. Bei Werten kleiner als 1 oder größer als 95 ertönt kein Tonsignal für das Zeilenende.

Die `MARGIN` Einstellung bleibt so lange wirksam, bis ein neuer `MARGIN` Wert eingegeben wird.

Programmierhinweis: Der Befehl `MARGIN` definiert auch die Anzahl Zeichen, die als Antwort auf eine `INPUT` Anweisung eingegeben werden können, bevor das Tonsignal ertönt.

Der Zeichensatz des HP-75

Den 256 Zeichen des HP-75 sind Zahlen, *Dezimalcodes*, zwischen 0 und 255 zugeordnet. Anhang D enthält eine Tabelle aller Zeichen und der ihnen zugeordneten Dezimalcodes. Fünfundneunzig dieser Zeichen (Dezimalcodes 32 bis 126) sind nach der ASCII-Konvention definierte Standard-Druckzeichen. Die Zeichen 0 bis 31 und 127 bis 255 können von anderen Computern und von HP-IL Einheiten anders interpretiert werden, als sie vom HP-75 angezeigt werden. Schlagen Sie gegebenenfalls im entsprechenden Handbuch Ihres Computers oder Peripheriegerätes nach.

Zeichen und Dezimalcodes (CHR$, NUM)

Die Funktion CHR$ gibt das entsprechende Anzeigezeichen eines gegebenen Dezimalcodes zurück.

Beispiel:

```
>chr$(65)
```

```
A
```

Großbuchstaben entsprechen den Dezimalcodes 65 bis 90.

CHR$ rundet numerische Ausdrücke auf ganzzahlige Werte und wandelt ganze Zahlen modulo 256 in eine Zahl im richtigen Bereich um.

Die Funktion NUM ist die Umkehrung von CHR$. Sie gibt den Dezimalcode eines gegebenen Zeichens zurück.

Beispiel:

```
>num('A')
```

```
65
```

Das Zeichen muß in Anführungszeichen eingeschlossen sein.

Der Dezimalcode von A.

NUM kann auf beliebigen Stringausdrücken operieren und gibt den Dezimalcode des *ersten* Zeichens des Strings zurück.

Beispiel:

```
>num(' ABC')
```

```
32
```

Gibt den Dezimalcode des ersten Zeichens im String, des Leerzeichens, zurück.

Die Zeichen mit den Dezimalcodes 128 bis 255 erscheinen unterstrichen in der Anzeige. Benutzen Sie den Dezimalcode des nicht unterstrichenen Zeichens, die Konstante 128 und die Funktion CHR$, um ein unterstrichenes Zeichen anzuzeigen. Um zum Beispiel A anzuzeigen:

```
>chr$(65+128)
```

```
A
```

65 steht für A.

Gibt das unterstrichene Zeichen zurück.

Eine andere Methode benutzt die Funktion NUM:

```
>chr$(num('A')+128)
```

```
A
```

NUM liefert den Dezimalcode von A.

Das gleiche Ergebnis.

Zeichenanzeige-Tastenfolge (SHIFT I/R)

Sie können mit 194 Tasten und Tastenkombinationen Anzeigezeichen erzeugen. Auch den Editier- und Systemtasten sind wie den Schreibmaschinentasten Anzeigezeichen zugeordnet. Sie können die meisten dieser Zeichen jedoch nicht einfach durch Drücken dieser Tasten anzeigen – viele Tasten führen vorbestimmte Operationen aus, wenn sie gedrückt werden. Sie können aber mit SHIFT I/R, der *Zeichenanzeige-Tastenfolge*, auf diese Zeichen zugreifen.

Der Taste APPT zum Beispiel ist das Zeichen ⧖ zugeordnet. Wenn Sie jedoch APPT allein drücken, schaltet der Rechner in den APPT- Modus. Um das Zeichen ⧖ anzuzeigen, müssen Sie zuerst SHIFT I/R betätigen. Dabei bleibt der Cursor in seiner jetzigen Position, und der HP-75 zeigt das Zeichen der nächsten Taste, die gedrückt wird, an. Drücken Sie SHIFT I/R und dann die Taste APPT: das Zeichen ⧖ erscheint.

Wenn Sie SHIFT I/R drücken und wieder loslassen, wird danach immer das Zeichen der nächsten Taste oder Tastenfolge angezeigt. Versuchen Sie eine Tastenfolge: Drücken Sie CTL 1 allein. Keine Reaktion. Nun drücken Sie SHIFT I/R und dann CTL 1 – Sie erhalten eine unterstrichene 1 in der Anzeige.

Wenn Sie das numerische Tastenfeld benutzen (CTL LOCK), werden nach SHIFT I/R die normalen Anzeigefunktionen dieser Tasten ausgeführt. Das U des numerischen Tastenfeldes zum Beispiel erzeugt eine 4. Wenn Sie SHIFT I/R und dann U drücken, wird das kleine u angezeigt.

SHIFT I/R ignoriert auch zuvor definierte Tastenbelegungen. Wenn Sie die nicht umgeschaltete Taste Q zur Anzeige Ihres Namens umdefiniert haben, dann zeigt SHIFT I/R, gefolgt von Q das gewöhnliche q an.

Was passiert, wenn Sie SHIFT I/R zweimal nacheinander drücken? Konsequenterweise erscheint das SHIFT I/R zugeordnete Zeichen (ç) in der Anzeige.

Alle Tasten und Tastenkombinationen des HP-75 besitzen zugeordnete Dezimalcodes. Der Taste APPT zum Beispiel entspricht der Dezimalcode 130. Mit den Funktionen SHIFT I/R und NUM können Sie den Dezimalcode einer bestimmten Taste oder Tastenkombination bestimmen. Um den Dezimalcode von APPT zu erhalten:

```
>num('⧖')
```

Sie müssen zuerst SHIFT I/R drücken, um das Anzeigezeichen von APPT zu erhalten.

```
 130
```

Der Taste APPT entspricht der Dezimalcode 130.

Der Zeichensatz in Anhang D enthält die Dezimalcodes des Tastenfelds und zeigt auf, welchen Tastenfolgen ein SHIFT I/R vorangehen muß, um die jeweiligen Zeichen anzuzeigen.

Informationen für den fortgeschrittenen Benutzer

Kontrollcodes (CTL BACK, CTL H, CTL M, CTL J)

Der HP-75 verfügt über 33 ASCII-definierte *Kontrollcodes*, die zur Steuerung des Informationsaustauschs zwischen angeschlossenen Peripheriegeräten dienen. Den Kontrollzeichen sind die Dezimalcodes 0 bis 31 und 127 zugeordnet. Sie können Kontrollzeichen entweder mit der Taste CTL oder mit der Funktion CHR$ erzeugen. Zum Beispiel kann das Kontrollzeichen «Null» sowohl mit CTL Leertaste als auch mit CHR$(0) erzeugt werden, das auf dem HP-75 als ◬ angezeigt wird.

Einige Kontrollzeichen besitzen eine spezielle Bedeutung für die Anzeige und für Peripheriegeräte des HP-75. Speziell das Zeichen *Escape* besitzt eine Vielzahl von Kontrollanwendungen. Das Escapezeichen wird entweder mit der Tastenfolge CTL BACK – wie durch die Aufschrift ESC über der Taste BACK angedeutet – oder mit CHR$(27) erzeugt. Der HP-75 zeigt das Escapezeichen nicht an, stattdessen zeigt er › und schaltet den Cursor aus. Der Cursor erscheint bei der Eingabe des nächsten Zeichens wieder. Die Reaktionen der Anzeige und der HP-IL Geräte auf Escapezeichen werden in Abschnitt 9 weiter erläutert.

Drei weitere spezielle Kontrollzeichen sind:

- Das *Backspace*-Zeichen (Dezimalcode 8) setzt den Cursor um eine Stelle zurück. Backspace kann durch Eintasten einer Zeichensequenz und anschließendes Drücken von [SHIFT] [I/R], gefolgt von [CTL] [H], [BACK] oder [SHIFT] [BACK], «angezeigt» werden. (Die drei Tastenfolgen sind identisch.) Beachten Sie, daß das Backspacezeichen den Cursor bewegt, ohne irgendwelche Zeichen zu löschen.
- Das Zeichen *Wagenrücklauf* (Dezimalcode 13) setzt den Cursor an den Anfang der Anzeige. Ein Wagenrücklauf kann durch Eingabe von [SHIFT] [I/R] und danach [CTL] [M] oder [RTN] «angezeigt» werden. Beachten Sie, daß bei einem Wagenrücklauf der Cursor bewegt wird, ohne daß der Zeileninhalt ausgeführt wird. (Lesen Sie dazu den folgenden Absatz «Wagenrücklauf/Zeilenvorschubtasten».)
- Das Zeichen *Zeilenvorschub* (Dezimalcode 10) setzt den HP-75 auf eine neue Anzeigezeile. Ein Zeilenvorschub (oder auch Line Feed) kann mit [CTL] [J] «angezeigt» werden.

Das folgende LINEFEED-Programm demonstriert die Wirkung von Backspace-, Wagenrücklauf-, Zeilenvorschub- und Escapezeichen auf die Anzeige des HP-75. Bereiten Sie den Rechner mit der Tastenfolge `edit 'linefeed', basic` [RTN] auf die Eingabe des Programms vor:

```
LINEFEED    B     0 11:50 02/05/83
```

Der Programmkopf. Drücken Sie [ATTN] und geben Sie das Programm LINEFEED ein.

```
1 input a
2 disp'Anfang';chr$(a);'Ende'
```

Das Programm besteht aus 2 Zeilen. Drücken Sie [RTN] nach jeder Zeile.

Lassen Sie das Programm viermal ablaufen. Geben Sie jedesmal, wenn die Eingabeaufforderung ? erscheint, einen neuen Dezimalcode – 8, 10, 13 oder 27 – ein, und drücken Sie [RTN].

Beispiele:

```
?8
                          PRGM
```

Spezifiziert das Backspacezeichen.

```
AnfanEnde
```

Der zweite String wird von der Position des zurückgesetzten Cursors aus angezeigt.

```
?27
                          PRGM
```

```
nde
```

Das Escapezeichen und der Buchstabe E bewirken ein vollständiges Löschen der Anzeige.

Die Funktion CHR$ wird oft in Programmen benutzt, um Kontrollzeichen an HP-IL Einheiten zu senden.

Wagenrücklauf/Zeilenvorschubtasten

Neun Tasten des HP-75 senden ein Wagenrücklauf- und ein Zeilenvorschubzeichen an die Anzeige und andere HP-IL Anzeigegeräte. Dazu gehören die sieben Systemtasten ([ATTN], [RTN], [TIME], [APPT], [EDIT], [FET], [RUN]) und zwei Editiertasten ([↑], [↓]). (Beachten Sie, daß [CTL] [M] die gleiche Funktion hat wie [RTN]).

Wenn Sie eine dieser Tasten drücken, wird vor der Funktionsausführung ein Wagenrücklauf/Zeilenvorschub ausgelöst. Bevor zum Beispiel die Taste [APPT] eine APPT-Maske erzeugt, wird der Cursor an den Anfang der Zeile gesetzt, und der HP-75 zeigt eine neue Zeile an. In den meisten Fällen werden Wagenrücklauf/Zeilenvorschub so schnell durchgeführt, daß sie nur bei Peripheriegeräten bemerkbar sind.

Abschnitt 3

Editieren von Files

Inhalt

Einführung

Das System zur Fileverwaltung des HP-75 ist sehr vielseitig und ermöglicht Ihnen, mehrere Programme, Textmemos, Terminkalender, umdefinierte Tastenfunktionen und andere Informationsblöcke zu speichern und zu bearbeiten. Ein File ist ein Speicherbereich, der über einen Namen aufgerufen und als Einheit behandelt werden kann. Ein File besteht aus einer Anzahl von Zeilen, die entweder über das Tastenfeld eingegeben oder von einem Massenspeichermedium kopiert wurden. Hier sind vier Beispiele für Anzeigezeilen aus Abschnitt 1, die in Files abgespeichert worden sind.

```
SAT 02/05/83 09:25 AM #3N !Anruf
```

In einem APPT-File mit dem Namen `appt` gespeichert.

```
>1 BEEP RND*999,RND*.5 @ GOTO 1
```

In einem BASIC-Programmfile mit dem Namen `SINGSONG` gespeichert.

```
>def key 'q','Ihr Name';
```

In einem speziell für Tastendefinitionen reservierten File mit dem Namen `keys` abgespeichert.

```
:10 An Chef
```

In einem Textfile namens NOTCHEF gespeichert.

Sie haben die Zeile in jedem Fall mit der Taste RTN in den File eingegeben. Die Taste RTN besitzt zwei wichtige Anwendungen in diesem Abschnitt:

- Eintragen neuer Zeilen in Files.
- Ausführen von Befehlen zur Filekontrolle, wie RENAME.

Filenamen

Jeder File im Speicher verfügt über einen eigenen Namen, den *Filenamen*. Es kann zum Beispiel nur ein File mit dem Namen FINANZ existieren. Ein Filename kann aus einem bis zu acht Zeichen bestehen. Das erste Zeichen eines Filenamens muß ein Buchstabe oder ein Punkt sein, und die verbleibenden Zeichen können eine beliebige Kombination aus Ziffern und Buchstaben sein. Beispiele zulässiger Filenamen sind C, ACC1, .TEMP, REV3, .92 und LOWSCORE. Sie können Filenamen in Kleinbuchstaben eingeben; diese werden vom HP-75 automatisch in Großbuchstaben umgewandelt. Ein Leerzeichen beendet den Filenamen, Folgezeichen werden ignoriert. Schließen Sie bei der Eingabe der Befehle dieses Abschnitts Filenamen in einfache oder doppelte Anführungszeichen ein. (Dazu gibt es zwei Ausnahmen: die Namen der Sonderfiles appt und keys sind als Kleinbuchstaben gespeichert und müssen ohne Anführungszeichen eingegeben werden.)

Ein mit einem Punkt beginnender Filename bezeichnet einen *Volativfile* (*flüchtigen* File). Wenn sich der HP-75 ausschaltet oder ausgeschaltet wird, werden alle Volativfiles beim nächsten Einschalten gelöscht. Volativfiles sind im wesentlichen temporäre Files, die nur für die momentane Arbeitsperiode benötigt werden.

Files im Speicher

Files werden in der Reihenfolge ihrer Erzeugung im Speicher abgelegt; der erste File befindet sich am «Boden». Sie sehen hier die relativen Lagen der in den Abschnitten 1 und 2 erzeugten Files. Die Abbildung stellt den Speicher des HP-75 dar und enthält Namen, Arten, Größen und das jeweilige Datum der dort gespeicherten Files.

```
Verfügbarer Speicher

LINEFEED    B    56 11:50 02/05/83
  Der zuletzt eingegebene File.
NOTCHEF     T    90 10:16 02/05/83
  Ein Textfile.
FINANZ      B  2248 13:52 07/13/82
  Obwohl die Anzeige das Aufzeichnungsdatum
  des Programms enthält, ist der File entsprechend
  der Zeit des Eintrags in den Speicher abgelegt.
SINGSONG    B    47 09:45 02/05/83
  Das erste Programm, das Sie gespeichert haben.
appt        A    28 09:30 02/05/83
  Ihr erster File, bei der ersten Terminplanung
  erzeugt.
```

Schlagen Sie den Speicherbedarf von Files in Anhang B nach.

Dieser BASIC-File wurde erzeugt,falls Sie das Programm auf Seite 43 eingegeben haben.

Der File keys war hier abgelegt, bis Sie ihn mit purge keys RTN gelöscht haben.

Der HP-75 kann soviele Files aufnehmen, wie der Speicherplatz erlaubt. Sechs Filetypen sind möglich:

- BASIC- oder Programmfiles, durch den Buchstaben B gekennzeichnet. BASIC Files bestehen aus fortlaufend numerierten Zeilen mit Programmanweisungen. Ein *privater* BASIC-File wird durch PB gekennzeichnet; dieses Programm kann zwar ausgeführt, aber nicht gelistet, geändert oder kopiert werden (Abschnitt 8, Seite 117).
- Textfiles, durch T gekennzeichnet. Textfiles sind numerierte Zeilen mit freien Zeichenfolgen.
- Terminfiles, durch A gekennzeichnet (Abschnitt 7, Seite 102).
- Tastenfiles, ebenfalls durch T gekennzeichnet. Tastenfiles sind spezielle Textfiles (Abschnitt 10, Seite 144).
- *Language Extension Files*, oder LEX-Files, durch L gekennzeichnet. LEX-Files sind spezielle Programmfiles, die auf vorbespielten Magnetkarten und -bändern sowie Einschub-ROMs verfügbar sind (Anhang B, Seite 277).
- *Logical Interchange Files*, oder LIF1-Files, durch I gekennzeichnet. LIF1-Files sind speziell formatiert für den Austausch von Information zwischen dem HP-75 und anderen Computern (Anhang B, Seite 274).

Dieser Abschnitt behandelt der Einfachheit halber Beispiele mit *Text*files, aber die Vorgehensweise bei der Zusammenstellung und Organisation von BASIC-Files ist die gleiche.

EDIT-Modus (EDIT)

Text- und BASIC-Files werden im EDIT-Modus erzeugt und modifiziert. Drücken Sie EDIT, um den EDIT-Modus zu setzen:

```
>
```

Der BASIC-Cursor und die Eingabeaufforderung erscheinen.

Wenn stattdessen die Text-Eingabeaufforderung (:) erscheint, geben Sie `purge` RTN ein, um den derzeitigen Textarbeitsspeicher zu löschen, und geben Sie dann `edit basic` RTN ein, um den HP-75 auf die Editierung eines BASIC-Files vorzubereiten. Die BASIC- Eingabeaufforderung erscheint bei Beginn der Eingabe oder wenn Sie ATTN drücken.

Die BASIC-Eingabeaufforderung meldet, daß der HP-75 zur Aufnahme von Programmanweisungen bereit ist. Die Text-Eingabeaufforderung meldet, daß der Computer zur Aufnahme von Textzeilen bereit ist. Hier sehen Sie die verschiedenen Operationen im EDIT-Modus:

BASIC-Eingabeaufforderung (>) in der Anzeige:

Erzeugen und Bearbeiten von BASIC Files.

```
100 DISP A;B;C
```

Beliebige Eingabeaufforderung: Berechnen von Ausdrücken.

```
sqr (2)
```

Ausführen von Systembefehlen.

```
delay 1
run 'finanz'
```

Text-Eingabeaufforderung (:) in der Anzeige:

Erzeugen und Bearbeiten von Textfiles.

```
10 Lieber Wolfgang,
```

Sie werden bemerken, daß Sie Systembefehle (wie `DELAY` und `RUN`) unabhängig von der Eingabeaufforderung in der Anzeige ausführen können. Der Unterschied besteht darin, daß BASIC-Zeilen bei der Eingabe in Maschinensprache umgewandelt werden, um die Programmausführung zu beschleunigen, während Textzeilen in der eingegebenen Form gespeichert werden. Deshalb können Textfiles nicht als Programme ausgeführt werden, auch wenn sie gültige BASIC-Anweisungen enthalten. (Der Befehl `TRANSFORM` (siehe Anhang B, Seite 274) ermöglicht Ihnen jedoch die Umwandlung von Textfiles in BASIC-Files und umgekehrt.)

Verfügbarer Speicherplatz (MEM)

Mit der Funktion MEM können Sie die Größe des derzeit verfügbaren Speicherraums prüfen.

Beispiel:

```
>mem
```

```
4912
```

Typisches Ergebnis: der HP-75 kann weitere 4912 Bytes an Information speichern.

Die Anzahl Files im Speicher und die Anzahl der Zeilen in einem File sind nur durch den verfügbaren Speicherraum beschränkt. Eine 30-Zeilen-Notiz zum Beispiel (ungefähr eine Schreibmaschinenseite) benötigt etwa 2500 Bytes Speicherplatz; das entspricht zwei beschriebenen Magnetkarten.

Die von MEM gefundene Anzahl Bytes kann davon abhängen, wann die Funktion benutzt wird.

Beispiel:

```
>mem;mem;mem
```

```
4912  4906  4900
```

Jede Ausführung der Funktion belegt kurzfristig sechs Bytes des Speichers.

Sie können mit MEM für jede Operation des HP-75 feststellen, wieviel Speicherplatz sie benötigt. Führen Sie MEM vor und nach einer Programm- oder Tastenfeldoperation durch, um deren Auswirkung auf den verfügbaren Speicherplatz festzustellen.

Beachten Sie, daß das Betriebssystem etwa 2100 Bytes des Speichers für Systemroutinen benötigt. Nach einem Zurücksetzen des Systems stellt der HP-75C dem Benutzer etwa 14.000 Bytes zur Verfügung. Dieser Benutzerspeicher kann durch Einsetzen eines Speichermoduls HP82700A um 8K (8192) Bytes vergrößert werden.

Erzeugen und Editieren von Files (CAT, EDIT, FETCH)

Im EDIT-Modus greift der HP-75 immer auf den *momentanen* File, einen BASIC- oder Textfile zu. Der momentane File ist immer der File, mit dem Sie gerade arbeiten. Sie können den momentanen File mit verschiedenen Befehlen wechseln. Zur Zeit ist der HP-75 «in» einem BASIC-File, wie Sie an der Eingabeaufforderung > sehen können. Führen Sie den Befehl CAT (*catalog*) aus, um mehr Information über den momentanen File anzuzeigen.

```
>cat
```

```
LINEFEED  B  56 11:50 02/05/83
```

Zeigt den Katalogeintrag des momentanen Files an.

Ein Katalogeintrag enthält:

- Den Namen des derzeit editierten Files (hier: LINEFEED).
- Den Filetyp (B für BASIC).
- Den vom File belegten Speicherplatz in Bytes. Anhaltspunkt: jedes Anzeigezeichen eines Textfiles benötigt ein Byte Speicherplatz. Eine 0 kennzeichnet einen leeren File.
- Die Uhrzeit der Erzeugung des Files.
- Das Datum der Erzeugung des Files.

Drücken Sie [→], um die Zeile nach links zu schieben; drücken Sie [←], um die Zeile nach rechts zu schieben. Mit [SHIFT] oder [CTL] können Sie die Wirkung von [→] und [←] modifizieren. Jede andere Eingabe bringt die Eingabeaufforderung und den Cursor in die Anzeige:

```
>
```

Der Katalogeintrag verschwindet.

Benutzen Sie den Befehl EDIT, um einen neuen Text- oder BASIC-File zu erzeugen. Der Befehl EDIT kann vier Formen annehmen:

EDIT '*Filename*', BASIC	Erzeugt einen BASIC-File des gegebenen Namens.
EDIT '*Filename*', TEXT	Erzeugt einen Textfile des gegebenen Namens.
EDIT BASIC	Erzeugt einen temporären BASIC-File mit dem Namen workfile (Arbeitsfile).
EDIT TEXT	Erzeugt einen temporären Textfile mit dem Namen workfile.

Auf den folgenden Seiten wird an einem Beispiel gezeigt, wie ein Textfile erzeugt, beschrieben, editiert und manipuliert wird. Geben Sie die folgende Sequenz ein, und drücken Sie [RTN]:

```
>edit 'beispiel',text
```

Erzeugt einen Textfile. Der Filename kann in einfache oder doppelte Anführungszeichen eingeschlossen werden.

```
BEISPIEL    T    0 13:07 02/05/83
```

Zeigt den Katalogeintrag des Files.

Wenn Sie [ATTN] drücken, erscheint die Texteingabeaufforderung (:) und zeigt die Bereitschaft zur Editierung eines Textfiles an. Geben Sie nun diese Zeichenfolge in die Anzeigezeile, die mit einer Zeilennummer beginnt, ein, und drücken Sie dann [RTN]:

```
:0Dies ist Zeile Null.
```

Eingabe einer Zeile in den File Beispiel.

```
:
```

Zeile 0 ist abgespeichert.

Zeile 0 ist noch immer die momentane Zeile des Files, obwohl sie nicht mehr angezeigt wird. Der HP-75 behält mit Hilfe eines *Filepointers* die Übersicht über die momentanen Zeilen eines Text- oder BASIC-Files. Drücken Sie nacheinander die Tasten [FET] und [RTN], um die momentane Zeile in die Anzeige zu holen (oder abzurufen). Mit [FET] wird das Wort FETCH in die Anzeige gebracht:

```
:FETCH
```

Die Taste [FET] ist eine Eingabehilfe. Sie könnten stattdessen auch das Wort fetch eingeben.

Führen Sie dann mit [RTN] den Befehl FETCH aus:

```
0Dies ist Zeile Null.
```

Die momentane Zeile erscheint genauso, wie sie eingegeben wurde.

Die momentane Zeile, d.h. die Zeile auf die der Filepointer deutet, wird auch als *anstehende* Zeile bezeichnet. *Drücken Sie [FET] und dann [RTN], wann immer Sie die anstehende Zeile zur Anzeige bringen wollen.* Die momentane Zeile bleibt fixiert, solange Sie den Filepointer nicht bewegen. Verlassen Sie zum Beispiel vorübergehend den EDIT-Modus kurzfristig, indem Sie [APPT] drücken:

```
Day Mo/Dy/Yr Hr:Mn AM 1N !Note
```

Mit [APPT] wird der HP-75 auf APPT-Modus geschaltet. Drücken Sie nun wieder [EDIT].

```
:
```

Zurück im EDIT-Modus.

Stellen Sie mit dem Befehl CAT fest, daß BEISPIEL der momentane File ist.

```
:cat
```

```
BEISPIEL    T    38 13:07 02/05/83
```

Der Filepointer ist immer noch auf dem File BEISPIEL positioniert. Der File ist auf 38 Bytes angewachsen.

Abrufen der momentanen Zeile mit FETCH bestätigt, daß sie die Nummer 0 besitzt.

```
:FETCH
```

```
0Dies ist Zeile Null
```

Die momentane Zeile des BEISPIEL-File.

Der Filepointer bleibt auch beim Ausschalten des HP-75 auf den momentanen Text- oder BASIC-File positioniert. Wenn Sie den Filepointer auf einen anderen File positionieren wollen, müssen Sie dazu den Befehl EDIT benutzen. Die Befehle CAT, FETCH und EDIT werden durchgehend in diesem Abschnitt verwendet.

Prüfen des Systemkatalogs (CAT ALL, EDIT)

Mit der Eingabe von cat RTN können Sie den Katalogeintrag des momentanen Text- oder BASIC-Files zur Anzeige bringen. Der Befehl CAT besitzt vier weitere Formen:

CAT ALL	Greift auf den gesamten Systemkatalog zu.
CAT '*Filename*'	Zeigt den Katalogeintrag des spezifizierten Files an.
CAT APPT	Zeigt den Katalogeintrag des Files appt an.
CAT KEYS	Zeigt den Katalogeintrag des Files keys an.

Beachten Sie, daß für die Files appt und keys keine Anführungszeichen verwendet werden.

Mit CAT ALL können Sie den Katalog aller im Speicher enthaltenen Files anzeigen. Drücken Sie CLR zum Löschen der Zeile und geben Sie ein:

```
>catall
```

```
Name   Type Len   Time   Date
```

Die erste Zeile der cat all-Liste wird mit der vorgegebenen Verzögerungszeit angezeigt.

Die erste Zeile zeigt den Inhalt der fünf Felder im Katalogeintrag. Nach einer der derzeitigen Verzögerungsrate entsprechenden Zeit wird der Katalogeintrag des zuletzt erzeugten Files angezeigt:

```
BEISPIEL    T    38 13:07 02/05/83
```

Beispiel, der momentane File, ist auch der zuletzt erzeugte File.

Mit dem Abwärtspfeil ↓ können Sie die Katalogeinträge der andern Files im Speicher zur Anzeige bringen. Die Katalogeinträge erscheinen in der Reihenfolge der Erzeugung der Files. Drücken Sie wiederholt die Taste ↓, bis Sie den ältesten File, appt, erreichen:

```
appt        A    28 09:30 02/05/83
```

Der älteste File im Speicher erscheint bei einer Auflistung mit CAT ALL zuletzt.

Mit dem Aufwärtspfeil (↑) können Sie die Katalogeinträge neuerer Files im Speicher betrachten. Mit SHIFT ↓ erhalten Sie den Katalogeintrag des ältesten Files; mit SHIFT ↑ den Eintrag des jüngsten Files.

Während einer Auflistung mit CAT ALL erhält die Taste [EDIT] eine neue Bedeutung. Wenn Sie [EDIT] drücken, wird der File, dessen Katalogeintrag in der Anzeige steht, zum momentanen File. (Wenn jedoch ein unbenannter, nichtleerer Arbeitsfile existiert, können Sie auf diese Weise auf keinen anderen File zugreifen; siehe Seite 63.)

Beispiel:

```
FINANZ     B 2248 13:52 07/13/82
```

Der Katalogeintrag des Programms FINANZ während CAT ALL. Drücken Sie [EDIT].

```
>
```

Der Filepointer zeigt auf die erste Zeile des Files, und der Cursor und die BASIC-Eingabeaufforderung erscheinen.

Beachten Sie, daß der File ein zur Editierung geeigneter BASIC- oder Textfile sein muß; andernfalls erhalten Sie die Warnung 65 – access restricted – und den Katalogeintrag dieses Files zurück, wenn Sie [EDIT] drücken.

Um zum Textfile BEISPIEL zurückzukehren, geben Sie entweder edit 'beispiel' [RTN] ein, oder führen Sie einen weiteren CAT ALL Befehl aus, und drücken Sie EDIT, wenn der Katalogeintrag für den File BEISPIEL in der Anzeige steht.

Löschen von Files (PURGE)

Der Befehl PURGE löscht einzelne Files aus dem Speicher.

PURGE	Löscht den momentanen File.
PURGE '*Filename*'	Löscht den spezifizierten File.
PURGE APPT	Löscht den File appt.
PURGE KEYS	Löscht den File keys.

Wenn Sie jetzt zum Beispiel purge [RTN] eingeben würden, würde der File BEISPIEL gelöscht werden.

Nach einem PURGE-Befehl wird der zuvor von diesem File belegte Speicherplatz wieder frei gegeben. Der Befehl PURGE kann als «gefährlich» eingestuft werden: er kann nicht mehr rückgängig gemacht werden. Wenn Sie den momentanen File löschen, erzeugt der HP-75 einen temporären Arbeitsfile des gleichen Typs und macht diesen zum momentanen File.

Eingabe von Zeilen in einen Textfile ([RTN])

Jede Zeile eines Textfiles beginnt mit einer Zeilennummer, einer ganzzahligen Zahl zwischen 0 und 9999, und besteht aus 96 Zeichenpositionen, einschließlich einer Stelle für die Eingabeaufforderung, einer bis zu vier Stellen für die Zeilennummer und der letzten Stelle für den Cursor. Damit kann jede Zeile aus bis zu 90-93 Zeichen bestehen. Drücken Sie nach der Eingabe einer Zeile [RTN]: der Inhalt der Anzeigezeile wird im momentanen File abgespeichert.

Geben Sie zum Beispiel ein:

```
:100 Beginnen Sie einfach zu schreiben, wenn Sie eine Zeile Text eingeben wollen.
```

Die Zeile beginnt beim 32. Zeichen im Anzeigefenster zu wandern.

Das Anzeigefenster am Ende der Eingabe.

Prüfen Sie Ihre Eingabe mit einer Kombination aus [←], [→] und den Vorwahltasten. Drücken Sie dann [RTN]:

```
:
```

Mit [RTN] wird die gesamte Zeile als Teil des momentanen Text-Files abgespeichert.

Holen Sie die anstehende Zeile – die Sie gerade eingegeben haben – zur Anzeige aus dem Speicher:

```
:FETCH
```

```
:100Beginnen Sie einfach zu sch
```

Die momentane Zeile. Der Cursor ist direkt hinter die Zeilennummer positioniert.

Löschen Sie die Anzeige mit CLR, ATTN oder EDIT. Dies hat auf die Speicherung im File keinen Einfluß.

Sie können die Zeilen in beliebiger Reihenfolge eingeben; sie werden aber entsprechend ihrer Zeilennummer, von der niedrigsten bis zur höchsten, angeordnet. Sie können nur immer in einen File, den momentanen File, Zeilen eingeben. *Sie müssen zur Eingabe einer Zeile in den File RTN drücken.*

Wenn Sie beim Editieren eines *Textfiles* RTN drücken, ignoriert der HP-75 alle führenden Leerzeichen und sucht nach einer gültigen Zeilennummer. Wenn er eine Zeilennummer findet, wird die Zeile wie sie ist im momentanen Textfile abgespeichert. `10+3` zum Beispiel wird als Zeile 10 mit dem Inhalt `+3` abgelegt. Groß- und Kleinbuchstaben, Leerzeichen nach der Zeilennummer, Anführungszeichen, Kontrollzeichen – kurz: alles – bleibt so erhalten, wie Sie es eingegeben haben.

Wenn Sie während der Editierung eines Textfiles Tastenfeldarithmetik durchführen wollen, dürfen Sie die Zeile nicht mit ganzen Zahlen zwischen 0 und 9999 beginnen. Beginnen Sie zum Beispiel mit einem Dezimalpunkt (.), einem Pluszeichen (+), einem Minuszeichen (−), einer linken Klammer (( ) oder einer einfachen numerischen Variablen (siehe Abschnitt 5, «Wertzuweisung auf Variablen»). Löschen Sie die Anzeige und geben Sie zum Beispiel ein:

```
:+10+3
```

Addition zweier Zahlen mit führendem Pluszeichen.

```
13
```

Das Ergebnis. Der Textfile bleibt von der Rechnung unbeeinflußt.

Automatische Zeilennumerierung (AUTO)

Wenn Sie den Befehl `AUTO` ausführen, werden die Zeilennummern eines Text- oder BASIC-File automatisch vorgegeben.

`AUTO` [*Anfangs-Zeilennummer* [, *Inkrementwert*]]

Löschen Sie die Anzeige und geben Sie ein:

```
:auto 1,1
```

```
:1
```

Beginnt die Numerierung bei Zeile 1. Nach der Zeilennummer wird ein Leerzeichen vorgegeben.

Geben Sie jetzt vier weitere Zeilen in den File `BEISPIEL` ein:

```
:1 Wer reitet so spaet
```

Beliebige Zeichenkombinationen können der Zeilennummer folgen.

```
:2
```

Zeilennummern werden in Einzelschritten erhöht, wie im Befehl `AUTO` spezifiziert wurde.

```
:2   durch Nacht und Wind,
```

Zwei weitere Leerstellen für den Versteil.

```
:3
```

Der Filepointer zeigt nach ihrer Numerierung auf die neue Zeile.

```
:3 es ist der Vater
```

```
:4
```

```
:4 mit seinem Kind...
```

```
:5
```

Die letzte Zeile des Fragments.

Beenden Sie hier die Numerierung mit [ATTN].

Der Filepointer zeigt auf die letzte Zeile, die Text enthält. Wenn Sie jetzt also FETCH ausführen, wird Zeile 4 in die Anzeige gerufen, obwohl die zuletzt angezeigte Zeile die Nummer 5 hatte.

Wenn Sie AUTO ohne Parameter ausführen, wird die Zeilennummer bei der momentanen Nummer *plus 10* fortgesetzt und in Zehnerschritten inkrementiert.

Beispiel:

```
:auto
```

```
:14
```

Beginnt die Numerierung bei Zeile 14, der momentanen Zeilennummer (4) plus 10.

Wenn Sie nur die Anfangszeilennummer spezifizieren, beginnt die Numerierung bei dieser Nummer und inkrementiert um jeweils 10. Geben Sie zum Beispiel ein, nachdem Sie [ATTN] gedrückt haben:

```
:auto 40
```

```
:40
```

Spezifiziert die Anfangsnummer.

Die erste angezeigte Zeile ist 40, dann folgt 50 usw. Drücken Sie einmal [RTN], um dies zu überprüfen.

Beenden Sie die automatische Zeilennumerierung mit [ATTN].

Beachten Sie, daß bei der automatischen Zeilennumerierung existierende Zeilen des Files nicht überschrieben werden. Weil zum Beispiel Zeile 4 im BEISPIEL-File existiert, wird mit auto 4 Zeile 4 abgerufen und nicht ersetzt. Mit anderen Worten, der Befehl AUTO holt Zeilen aus dem momentanen Text- oder BASIC-File.

Syntaxfehler

Beim Editieren eines Textfiles können Sie jeden Systembefehl (wie FETCH und CAT) direkt über das Tastenfeld eingeben. Sie können weiterhin BASIC-Anweisungen (wie BEEP) und Ausdrücke (wie SQR(2)) ausführen. Für alle Zeilen, die nur Text enthalten, sollten Sie jedoch Zeilennummern eingeben. Wenn Sie das nicht tun und [RTN] drücken, kann ein *Syntaxfehler*, ein sprachbezogener Fehler auftreten.

Der HP-75 reagiert wie folgt auf Syntaxfehler:

- Tonsignal (falls BEEP ON gesetzt).
- Setzen der Statusanzeige **ERROR**.
- Anzeige einer Fehlermeldung entsprechend der eingestellten Verzögerungszeit.
- Rückruf der fehlerhaften Zeile in die Anzeige.

Geben Sie zum Beispiel ein, nachdem Sie [ATTN] gedrückt haben:

```
:--Goethe--
```

```
:--Goethe--
      ERROR
```

Der Name des Poeten hat keine Zeilennummer erhalten.

Syntaxfehler.

Die Fehlerkorrektur ist einfach:

- Rufen Sie mit [SHIFT][FET] die Fehlermeldung ab. In diesem Fall bedeutet `missing parameter`, daß die Anzeigezeile nicht als Ausdruck interpretiert werden konnte.
- Bestimmen Sie mit der Funktion `ERRN` (geben Sie `errn` [RTN] ein) die Fehlernummer.
- Löschen Sie mit [CLR], [ATTN] oder [EDIT] die Fehlerbedingung, die Fehler-Statusanzeige, und die Anzeigezeile.
- Korrigieren Sie den Fehler mit den Editiertasten. Drücken Sie in diesem Fall [I/R], dann [SHIFT], geben Sie eine Zeilennummer und ein Leerzeichen ein und drücken Sie dann [RTN]:

```
:34 --Goethe--
      ERROR
```

Eingabe einer Zeilennummer und eines Leerzeichens mit dem Einführungscursor.

```
:
```

Die korrigierte Zeile wird im File als Zeile 34 gespeichert.

Erinnern Sie sich, daß [CTL][FET] den Inhalt des Eingabebuffers abruft. Wenn Sie Zeilen in einen File einfügen, wird mit [CTL][FET] die zuletzt eingegebene Zeile angezeigt.

Beispiel:

```
:34 --Goethe--
```

[CTL][FET] ruft die Zeichen ab, wie sie beim Drücken von [RTN] eingegeben waren.

Editieroperationen

Sie können immer, wenn der HP-75 im EDIT-Modus ist, den momentanen File editieren – Sie können Zeilen anhängen, prüfen, einfügen, verschieben, listen, umnumerieren und löschen.

Mit den in Abschnitt 2 besprochenen Editiertasten und -Tastenfolgen können Sie *einzelne* Zeilen korrigieren. Sie haben die Möglichkeit, die Anzeigezeile nach Ihren Wünschen zu ändern, bevor sie in den File eingetragen wird (d.h. bevor Sie [RTN] drücken).

Schrittweises Durchlaufen eines Files ([↑], [↓])

Die Tasten [↑] und [↓] bewegen den Filepointer im momentanen File auf- und abwärts. Wenn Sie [↑] drücken, erscheint die der momentanen Zeile vorhergehende Zeile editierbereit in der Anzeige, und entsprechend ist nach [↓] die der anstehenden Zeile folgende Zeile sichtbar. Beachten Sie, daß mit [↑] oder [↓] die angezeigte Zeile *nicht* im File gespeichert wird. Wenn Sie eine Zeile geändert haben und diese Änderung im File gespeichert haben wollen, müssen Sie [RTN] drücken, bevor Sie mit [↑] oder [↓] fortfahren.

Wenn Sie die Taste [↑] oder [↓] gedrückt halten, wird deren Tastenfunktion wiederholt. Mit [SHIFT][↑] wird der Pointer auf die erste Zeile des File gesetzt, nach [SHIFT][↓] zeigt er auf die letzte Zeile des File.

Abruf von Zeilen eines Files (`FETCH`)

Eine Anwendung des Befehls `FETCH` ist die Anzeige der anstehenden Zeile des momentanen Files. Wenn ein String in Anführungszeichen oder eine Zeilennummer oder beides spezifiziert wird, kann `FETCH` den Filepointer auf jede beliebige Zeile des Files positionieren. Hier sind die vier Formen des Befehls `FETCH`:

`FETCH`	Holt die anstehende Zeile.
`FETCH` *Zeilennummer*	Holt die spezifizierte Zeile.
`FETCH` '*Suchstring*'	Holt die nächste nach der anstehenden den *Suchstring* enthaltende Zeile.
`FETCH` '*Suchstring*' , *Zeilennummer*	Holt, beginnend bei der spezifizierten Zeile, die erste Zeile, die den *Suchstring* enthält.

Die Taste [FET] ist eine Eingabehilfe – Drücken von [FET] erzeugt das Wort FETCH in der Anzeige. Wenn Sie einen String in Anführungszeichen hinzufügen, wird nach diesem Suchstring *von der Zeile nach der anstehenden Zeile an* bis zum Fileende gesucht.

Beispiel: Suchen Sie im File BEISPIEL die Zeile, die die Zeichen acht enthält. Setzen Sie zuerst den Filepointer auf die erste Zeile des Files (mit [SHIFT] [↑]), und löschen Sie dann die Anzeige (mit [CLR]) und geben Sie ein:

```
:FETCH 'acht'
```

Der Befehl mit einem Ausdruck in Anführungszeichen. Die gesuchte Zeile muß *nach* der anstehenden Zeile auftreten.

```
:2   durch Nacht und Wind
```

Die erste Zeile nach der anstehenden Zeile, die den Ausdruck enthält. Der Cursor steht auf dem a.

Die Zeichen im FETCH String müssen genauso angeordnet sein wie in der gewünschten Zeile. Wenn die Zeile gefunden wird, steht der Cursor auf dem ersten Zeichen des angegebenen Strings.

Sie brauchen für den Befehl FETCH nur einen kurzen Ausdruck (wie 'cht' oder gar nur 'ht') anzugeben. Der Ausdruck kann in einfache oder doppelte Anführungszeichen eingeschlossen sein, solange sie nur gepaart sind. Wenn der Suchstring nach der anstehenden Zeile im File nicht auftritt, bleibt der Pointer nach dem Befehl FETCH bei der anstehenden Zeile stehen.

FETCH *Zeilennummer* holt die spezifizierte Zeile:

```
:FETCH 100
```

Der Befehl mit einer Zeilennummer.

```
:100 Beginnen Sie einfach zu suchen
```

Zeile 100 ist geholt und zum Editieren bereit.

Wenn diese Zeile nicht existiert, zeigt der HP-75 die spezifizierte Zeilennummer an; sie können die neue Zeile eingeben.

Beispiel:

```
:FETCH 102
```

Abrufen einer nichtexistenten Zeile.

```
:102
```

Die spezifizierte Nummer wird angezeigt.

Beachten Sie, daß die neue Zeile nicht im File abgespeichert wird, wenn Sie nicht mindestens ein Zeichen eingeben und [RTN] drücken. Andernfalls verschwindet die neue Zeile, sobald der Filepointer auf eine neue Zeile bewegt wird.

Wenn Sie einen Suchstring *und* eine Zeilennummer spezifizieren, beginnt der HP-75 bei der spezifizierten Zeile und sucht nach der ersten Zeile, die den spezifizierten String enthält.

Beispiel:

```
:FETCH 'spaet',0
```

Trennen Sie die zwei Parameter durch ein Komma.

```
:1 Wer reitet so spaet
```

Der HP-75 sucht von Zeile 0 an nach der ersten den String enthaltenden Zeile.

Korrigieren von Zeilen

Eine existierende Zeile in einem Textfile wird wie folgt modifiziert:

1. Rufen Sie mit FETCH, [↑] oder [↓] die Zeile in die Anzeige.
2. Ändern Sie die angezeigte Zeile.
3. Drücken Sie [RTN]. Die editierte Zeile ersetzt die ursprüngliche Zeile des Files.

Drücken Sie zum Beispiel SHIFT ↑:

```
:0Dies ist Zeile Null.
```

Mit SHIFT wird der Filepointer auf die erste Zeile des Files gesetzt.

Fügen Sie nun ein Leerzeichen zwischen Zeilennummer und dem ersten Zeichen (D) ein: Drücken Sie I/R, dann die Leertaste:

```
:0 Dies ist Zeile Null.
```

Editieren der anstehenden Zeile. Mit RTN wird die neue Version im File gespeichert.

Wenn Sie die Zeilen*nummer* durch eine neue Zeilennummer ersetzen und dann RTN drücken, wird die neue Zeile in den File eingetragen. Die alte Zeile bleibt jedoch – mit der alten Zeilennummer – ebenfalls im File. Die alte Zeile wird also dupliziert.

Einfügen von Zeilen

Eine neue Zeile wird wie folgt in einen Textfile eingefügt:

1. Geben Sie die neue Zeile mit einer geeeigneten Zeilennummer ein.
2. Drücken Sie RTN, um die neue Zeile in den File einzugeben.

Denken Sie daran, daß Zeilennummern ganzzahlig sein müssen. Es ist nicht möglich, eine neue Zeile zwischen zwei aufeinanderfolgende Zeilen, wie zum Beispiel zwischen die Zeilen 2 und 3, einzufügen. Sie können aber die Abstände zwischen Zeilennummern mit dem Befehl RENUM (Seite 57) ändern.

Verschieben von Zeilen

Sie können eine Zeile wie folgt im momentanen File verschieben:

1. Stellen Sie den Filepointer mit FETCH, ↑ oder ↓ auf die gewünschte Zeile.
2. Ändern Sie die Zeilennummer.
3. Drücken Sie RTN, um die neue Zeile im Speicher abzulegen.
4. Löschen Sie die Originalzeile, indem Sie eine leere Zeile mit der gleichen Nummer eingeben.

Befolgen Sie diese Vorgehensweise, um zum Beispiel Goethes Namen an den Kopf des Gedichtes zu stellen:

```
:FETCH 'Goe',0
```

Suchbeginn am Fileanfang.

```
:34 --Goethe--
```

Zeile 34 steht nun an.

```
: 0--Goethe--
```

Benutzen Sie SHIFT ←, ein Leerzeichen und eine Null zum Ändern der Zeilennummer.

```
:
```

Speichert diese Zeile im File; die ursprüngliche Zeile 0 wird ersetzt.

Löschen Sie schließlich die alte Zeile (34). Geben Sie ihre Zeilennummer ein und drücken Sie RTN:

```
:34
```

Die alte Zeilennummer.

```
:
```

Löscht Zeile 34.

Mit dieser Methode können Sie Zeilen im momentanen File beliebig umordnen. Mit dem Befehl DELETE (Seite 59) können Sie unerwünschte Zeilen ebenfalls löschen.

Beim Einfügen einer Zeile oder beim Löschen einer Zeile durch Eingabe der Zeilennummer und RTN wird der Filepointer auf diese Zeile positioniert. Als Sie zum Beispiel 34 RTN eingegeben haben, wurde diese Zeile zur anstehenden Zeile.

Listen von Zeilen (LIST, PLIST)

Mit dem Befehl FETCH und den Tasten ↑ und ↓ können Sie einzelne Zeilen aus dem momentanen File abrufen und für die Editierung verfügbar machen. Zusätzlich können Sie mit den Befehlen LIST und PLIST Zeilen eines Files zur Anzeige bringen. Mit LIST und PLIST können Sie weiterhin auf beliebige Text- und BASIC-Files im Speicher zugreifen und einzelne oder alle Zeilen anzeigen. Diese Zeilen können nur zur Prüfung gelistet, nicht aber editiert werden.

```
 LIST  [Anfangszeile[, Endzeile]]
LIST  'Filename'[, Anfangszeile[, Endzeile]]
```

```
PLIST  [Anfangszeile[, Endzeile]]
PLIST  'Filename'[, Anfangszeile[, Endzeile]]
```

Geben Sie nur LIST ein, wenn Sie alle Zeilen des momentanen Files listen wollen. Mit ATTN können Sie das Listen abbrechen. Geben Sie LIST RTN ein:

```
0 --Goethe--
1 Wer reitet so spaet
2   durch Nacht und Wind,
3 es ist der Vater
4 mit seinem Kind ...
40
100 Beginnen Sie einfach zu schr
eiben, wenn Sie eine Zeile Text
eingeben wollen.
```

Die Zeilen werden in der Reihenfolge ihrer Zeilennummern mit der derzeitigen Verzögerungszeit aufgelistet.

Zeile 40 besteht aus einer Zeilennummer und einem Leerzeichen. Zeile 100 wird in Gruppen zu 32 Zeichen angezeigt, wie mit WIDTH spezifiziert wurde.

Hinweise zu LIST:

- Während einer Auflistung bleibt der Pointer auf die anstehende Zeile positioniert.
- Bei einer WIDTH-Vorgabe größer als 32 wandern die Zeilen beim Listen über die Anzeige.
- Sie können die momentane Verzögerung umgehen, wenn Sie eine beliebige Taste außer ATTN, SHIFT oder CTL drücken. Der Listvorgang wird dann sofort bei der nächsten Zeile fortgesetzt.

Nach einem Listen von BEISPIEL zeigt die Anzeige:

```
eingeben wollen.
```

Beachten Sie, daß weder der Cursor noch die Eingabeaufforderung in der Anzeige stehen.

Gelistete Zeilen sind für die Editierung nicht verfügbar, obwohl die Tasten → und ← aktiv sind. Beim Drücken einer anderen Taste verschwindet die gelistete Zeile, und der Cursor und die Eingabeaufforderung erscheinen wieder. Der Filepointer bleibt auf die vor dem Listen anstehende Zeile gesetzt. Dies sehen Sie, wenn Sie FET und dann RTN drücken:

```
:FETCH ▮
```

Abrufen der anstehenden Zeile.

```
34 ▮
```

Zeile 34 ist die anstehende Zeile von zuvor. Sie ist leer, weil Sie sie gelöscht haben.

Mit LIST *Zeilennummer* wird die im Befehl spezifizierte Zeile angezeigt. Löschen Sie die Anzeige und geben Sie zum Beispiel ein:

```
:list 4
```

```
4  mit seinem Kind...
```

Die spezifizierte Zeile wird ohne Eingabeaufforderung und Cursor angezeigt.

Bei Angabe einer nichtexistenten Zeile wird der Befehl LIST ignoriert.

Mit dem Befehl LIST *Anfangszeile, Endzeile* können Sie alle Zeilen zwischen den und einschließlich den spezifizierten Zeilen auflisten.

Sie können auch Zeilen eines anderen Text- oder BASIC-Files auflisten. In den folgenden Beispielen werden jeweils eine oder mehrere Zeilen des Textfiles NOTCHEF aus Abschnitt 1 gelistet:

```
:list 'notchef'
```

Listet den gesamten File.

```
:list 'notchef', 20
```

Listet Zeile 20 des Files.

```
:list 'notchef',10,30
```

Listet die Zeilen 10 bis 30 einschließlich.

Der Befehl PLIST (*print list*) arbeitet wie der Befehl LIST, mit vier Unterschieden:

- Bei PLIST werden die Zeilen des spezifizierten Files auf die momentanen PRINTER IS Einheiten ausgegeben. Sind keine Druckereinheiten zugeordnet, dann wird der File auf die Anzeige des HP-75 und auf die DISPLAY IS Einheiten ausgegeben (Seite 128).
- Der DELAY-Wert beeinflußt die Druckgeschwindigkeit von externen Druckern nicht. Drucklisten auf externen Druckern werden ohne Verzögerungszeit zwischen den Zeilen erzeugt.
- Zur Modifikation der Anzahl der Zeichen pro Druckzeile ist anstelle von WIDTH der Befehl PWIDTH zu verwenden.
- PLIST listet die Zeilen eines Textfiles *ohne Zeilennummern*. (BASIC-Files werden vollständig mit ihren Zeilennummern ausgedruckt.)

PLIST ist nützlich, um Ihre Textfiles ohne Zeilennummern auszudrucken. Alle Leerzeichen nach Zeilennummern bleiben bei einer PLIST-Operation erhalten.

Umnumerierung von Zeilen (RENUMBER)

Der Befehl RENUMBER wird zur Umnumerierung einzelner oder aller Zeilen eines Text- oder BASIC-Files benutzt.

```
RENUMBER [Anfangszeile[ , Inkrementwert[ , von Zeilennummer alt
  [ , bis Zeilennummer alt]]]
```

RENUMBER ohne Spezifikationen bewirkt, daß der ganze File umnumeriert wird; die Zeilennummern beginnen dann mit 10 und werden jeweils um 10 inkrementiert. Wenn nur die *Anfangszeile* spezifiziert wird, beginnt die Umnumerierung bei dieser Zeile und wird um 10 inkrementiert. Beachten Sie, daß der ganze File umnumeriert wird, wenn Sie weniger als drei Parameter eingeben.

Beispiele: Drücken Sie nach jedem RENUMBER-Befehl [RTN], und prüfen Sie die Umnumerierung mit list [RTN] nach.

```
:renumber
```

Numeriert den momentanen File (BEISPIEL) um, so daß die erste Zeilennummer 10, die nächste 20 wird usw.

```
:renumber 100
```

Numeriert den ganzen File um. Die Anfangsnummer ist 100, das Inkrement ist 10.

```
:renumber 200,5
```

Numeriert den ganzen File um. Die Anfangsnummer ist, 200, das Inkrement ist 5.

Mit Hilfe der letzten beiden Parameter können Sie ausgewählte *Teile* des momentanen Files umnumerieren.

Beispiele:

```
:renumber 800,2,220
```

Numeriert die Zeilen 220 bis Fileende um (da kein Endparameter angegeben ist). Diese letzten Zeilen beginnen bei 800 und werden um 2 inkrementiert.

```
:renumber 10,1,1,210
```

Numeriert die Zeilen 1 bis 210 um; bei Zeile 10 beginnend und um 1 inkrementierend.

Wenn Sie die letzten drei Beispiele bearbeitet haben, erhalten Sie folgende Fileliste:

```
10 --Goethe--
11 Wer reitet so spaet
12   durch Nacht und Wind,
215 es ist der Vater
800 mit seinem Kind ...
802
804 Beginnen Sie einfach zu schr
eiben, wenn Sie eine Zeile Text
eingeben wollen.
```

Der umnumerierte Anfangsteil.

Der umnumerierte Endteil.

Der Filepointer bleibt während der Umnumerierung auf der anstehenden Zeile stehen, obwohl sich die Zeilennummer dieser Zeile eventuell ändert.

Der Befehl RENUMBER kann Filesegmente komprimieren oder expandieren oder offene Nischen im File erzeugen; er kann jedoch nicht die *Reihenfolge* der Zeilen verändern. Immer wenn eine der folgenden beiden Bedingungen auftritt, erhalten Sie Warnung 90 bad line number; und es wird eine Ersatznumerierung durchgeführt:

- Ein RENUMBER Befehl versucht, Zeilenfolgen umzuordnen oder zur Überlappung zu bringen.
- Ein RENUMBER Befehl erzwingt eine Zeilennummer über 9999.

In diesen Fällen numeriert das System den angegebenen Fileteil mit einem Inkrement 1, beginnend bei der kleinsten verfügbaren Zeilennummer, um.

Zusammen mit dem Befehl MERGE (Seite 60) ermöglicht Ihnen RENUMBER, zwei getrennte Files zu einem großen File zu kombinieren.

Löschen von Zeilen (DELETE)

Der Befehl DELETE ermöglicht Ihnen das Löschen unerwünschter Zeilen, ohne andere Zeilen zu beeinflussen.

```
DELETE [Anfangszeile [, Endzeile]]
```

DELETE allein löscht die anstehende Zeile; der Filepointer deutet jedoch weiterhin auf diese Zeile.

DELETE *Zeilennummer* löscht die spezifizierte Zeile.

Beispiel:

```
:delete 802
```

Mit RTN wird Zeile 802 gelöscht.

DELETE *Anfangszeile*, *Endzeile* löscht den Fileteil zwischen diesen beiden Zeilennummern einschließlich.

Beispiel:

```
:delete 12,800
```

Drücken von RTN löscht die Zeilen 12 bis 800 einschließlich.

Der Befehl DELETE 0,0000 hat die gleiche Wirkung wie das Löschen des gesamten Files, außer daß DELETE den Filenamen nicht zerstört. Der Filepointer wird vom Befehl DELETE nicht beeinflußt. Er bleibt auf der anstehenden Zeile stehen, selbst wenn diese gelöscht wird.

Erinnern Sie sich – die Taste DEL löscht Zeichen, nicht Zeilen.

Manipulation von Files

Die Befehle RENAME, MERGE, COPY, EDIT und NAME geben Ihnen weitere Möglichkeiten zur Bearbeitung von Files im Speicher.

Umbenennen von Files (RENAME)

Der Befehl RENAME ändert den Namen eines Files.

```
RENAME ['alter Filename'] TO 'neuer Filename'
```

Wenn nur der *neue Filename* spezifiziert wird, benennt der HP-75 den momentanen File um. Geben Sie die folgende Zeile ein und drücken Sie RTN:

```
:rename to 'lb'
```

Benennt den momentanen File (BEISPIEL) in LB um.

Mit CAT können Sie den neuen Filenamen anzeigen:

```
:cat
```

```
LB        T 132 13:07 02/05/83
```

Wenn Sie einen anderen File umbenennen wollen, geben Sie dessen derzeitigen Namen im RENAME-Befehl an.

Beispiel:

```
:rename 'notchef' to 'hart'
```

Mit RTN wird NOTCHEF in HART umbenannt.

Prüfen Sie den Namenswechsel mit `cat 'hart'` RTN nach. Schlagen Sie dazu auch die Diskussion des Befehls NAME nach (Seite 64).

Mischen von Files (MERGE)

Gelegentlich müssen Sie zwei Text- oder BASIC-Files zu einem einzigen File kombinieren. Dies wird durch den Befehl MERGE ermöglicht.

```
MERGE 'Filename' [,Anfangszeile [,Endzeile]]
```

Der Befehl MERGE fügt die spezifizierten Zeilen des spezifizierten Files in den *momentanen* File ein. Beide Files müssen vom gleichen Typ sein, und beide müssen sich im Speicher befinden. Der spezifizierte File bleibt erhalten.

Der Befehl MERGE allein kombiniert den *gesamten* spezifizierten File mit dem momentanen File. Wenn nur eine Zeilennummer spezifiziert ist, wird nur diese Zeile eingefügt. Wenn zwei Zeilennummern spezifiziert werden, werden alle Zeilen von der *Anfangszeile* bis zur *Endzeile* einschließlich eingefügt.

Beispiel:

```
:merge 'hart',20,40
```

Nach RTN werden die Zeilen 20 bis 40 des Textfiles HART in den momentanen File (LB) eingefügt.

Listen Sie danach den momentanen File, um die Mischoperation zu kontrollieren. Eine Abfrage des Systemkatalogs (CAT ALL) ergibt, daß der eingefügte File HART noch immer im Speicher existiert. Der Filepointer wird bei der Operation MERGE nicht bewegt.

> **Wichtig:** Während einer MERGE-Operation erfolgt keine Umnumerierung. Wenn der momentane File Zeilen mit gleichen Nummern wie Zeilen des eingefügten Teils besitzt, werden diese Zeilen des momentanen Files überschrieben. Sie müssen deshalb sicherstellen, daß die beiden Files sich nicht überlappen. Ändern Sie den Bereich der Zeilennummern des momentanen Files, falls notwendig, mit dem Befehl RENUMBER.

Kopieren von Files auf Magnetkarten (COPY)

Mit dem Befehl COPY, der in Abschnitt 1 zum Einlesen eines aufgezeichneten Programms in den Speicher benutzt wurde, können Sie auch einen File vom Speicher auf eine oder mehrere Magnetkarten schreiben.

```
COPY ['Filename'] TO CARD
```

Ohne Angabe eines Filenamens greift der Befehl COPY auf den momentanen File zu. Wenn Sie einen Filenamen spezifizieren, können Sie einen anderen File des Speichers kopieren. Geben Sie, um den momentanen File zu kopieren, folgendes ein:

```
:copy to card
```

```
1 track(s) needed
```

Diese Meldung wird nach den derzeitigen DELAY-Vorgaben angezeigt.

Nach der Anzeige dieser Meldung meldet der HP-75:

```
Copy to card; Align & [RTN]
```

Wählen Sie eine leere Magnetkarte aus Ihrem Benutzer-Paket, führen Sie sie so weit in den Eingabeschlitz ein, daß dessen Kante zwischen den beiden Markierungen steht, und drücken Sie RTN.

```
Pull card...
```

Ziehen Sie die Magnetkarte wie schon zuvor langsam und gleichmäßig durch den Kartenleser.

Danach sollten Sie sehen:

```
Verify card: Align & [RTN]
```

Der Kartenleser fordert einen zweiten Durchgang der selben Spur, um die eingelesene Information zu überprüfen.

Führen Sie die Magnetkarte nochmals ein und drücken Sie RTN:

```
Pull card...
```

Der HP-75 gibt Ihnen nach der Anforderung etwa 5 Sekunden zum Durchziehen der Magnetkarte.

Wenn der File fehlerlos kopiert wurde, erscheinen nach dem zweiten Durchgang der Cursor und die Eingabeaufforderung:

```
:
```

Zeigt an, daß der File auf die Magnetkarte kopiert worden ist.

Falls nach dem zweiten Durchgang `Warning: verify failed` angezeigt wird, bedeutet dies, daß die aufgezeichnete Information fehlerhaft war. In diesem Fall fordert Sie der HP-75 zu einem weiteren Kopiervorgang auf.

```
Copy to card: Align & [RTN]
```

Sie müssen die Spur noch zweimal durch den Kartenleser führen; einmal zum Kopieren und einmal zum Überprüfen.

Die Sauberkeit der Magnetkarte ist bei einem Kopiervorgang auf die Magnetkarte *von entscheidender Bedeutung*. Eine schmutzige oder beschädigte Magnetkarte kann zu unendlich oft wiederholten Warnungen und Kopieraufforderungen führen. Hinweise zur Reinigung der Magnetkarten finden Sie im Anhang B.

Folgen Sie der Anleitung zum Einlesen des Programms `FINANZ` in Abschnitt 1, wenn Sie einen File in den Speicher einlesen wollen. Dazu müssen Sie dem einzulesenden File einen neuen Namen geben – jeder File im Speicher muß einen eigenen Namen haben.

Duplizieren von Files (`COPY`)

Der Befehl `COPY` erlaubt Ihnen nicht nur, Files auf Magnetkarten zu kopieren und von Magnetkarten einzulesen, Sie können mit `COPY` auch Files im Speicher duplizieren.

```
COPY ['ursprünglicher Filename'] TO 'neuer Filename'
```

Wenn Sie einen File kopieren, erzeugen Sie einen neuen File und übertragen alle Zeilen des Originals. Ohne Spezifikation des *ursprünglichen Filenamens* wird der momentanen File dupliziert.

Beispiele:

```
:copy to 'neu'
```

Mit RTN wird der Inhalt des momentanen Files (`LB`) in den File `NEU` kopiert.

```
:copy 'singsong' to 'klang'
```

Mit RTN wird der Inhalt von SINGSONG in einen neuen File mit dem Namen KLANG dupliziert.

Prüfen Sie nun den Katalogeintrag von NEU:

```
:cat 'neu'
```

```
NEU      T 193 15:04 02/05/83
```

Sie sehen, daß NEU vom gleichen Typ und gleicher Größe ist wie der momentane File.

Beachten Sie, daß die im Katalogeintrag von NEU angegebene Uhrzeit von der Zeit des LB-File verschieden ist, weil der File NEU erst bei der Ausführung des Befehls COPY erzeugt wurde. Mit dem Befehl COPY können Sie eine Kopie eines Files anlegen, den Sie editieren wollen, oder das Datum und die Uhrzeit eines Files auf den neuesten Stand bringen.

Aufsuchen und Erzeugen von Files (EDIT)

Mit dem Befehl EDIT können Sie den Filepointer von einem File zum nächsten bewegen und neue Files erzeugen.

```
EDIT  ['Filename'  [,BASIC]
                   [,TEXT]

      BASIC
      TEXT
      KEYS
```

Wenn im Speicher ein File existiert, wird bei der Spezifikation dieses Files in einem EDIT Befehl der Filepointer auf die erste Zeile dieses Files gesetzt. Bewegen Sie den Filepointer zum Beispiel zur ersten (und einzigen) Zeile des Files SINGSONG:

```
:edit 'singsong'
```

```
SINGSONG    B    47 09:54 02/05/83
```

Der Katalogeintrag des Files SINGSONG wird angezeigt.

Drücken Sie FET, gefolgt von RTN, um die Pointerstellung zu prüfen. Da SINGSONG ein BASIC-File ist (wie das B im Katalogeintrag andeutet), erscheint die BASIC- anstelle der Text-Eingabeaufforderung. Der File ist editierbereit. Sie können Zeilen hinzufügen, listen, ändern, umnumerieren, löschen usw.

Erzeugen Sie neue BASIC- und Textfiles mit dem Befehl EDIT. Spezifizieren Sie jetzt zum Beispiel einen nichtexistierenden File:

```
>edit 'barbara'
```

```
BARBARA    B    0 15:40 02/05/83
```

Der File BARBARA, ein BASIC-File, wurde erzeugt.

Wenn weder BASIC noch TEXT im BASIC-Befehl spezifiziert wird, dann wird ein File gleichen *Typs* wie der momentane File erzeugt. Der File BARBARA zum Beispiel wurde als BASIC-File erzeugt – der gleiche Filetyp wie SINGSONG.

Sie können BASIC- oder TEXT-Files erzeugen, indem Sie im Befehl EDIT BASIC oder TEXT spezifizieren.

Beispiel:

```
>edit 'inge',text
```

Trennen Sie die beiden Parameter durch ein Komma.

```
INGE    T    0 15:42 02/05/83
```

Der spezifizierte Textfile wurde erzeugt.

Beachten Sie, daß Sie nicht den gleichen Filenamen für zwei verschiedene Files verwenden können, auch nicht, wenn es sich um verschiedene Filetypen handelt.

Der Systemkatalog enthält die Files in der Reihenfolge ihrer Erzeugung. INGE zum Beispiel, der neueste File, erscheint in einer CAT ALL-Auflistung zuerst. Um Speicherplatz zu sparen, wird jedoch ein leerer File im Speicher gelöscht, sobald Sie beginnen, einen anderen File zu editieren – und erscheint dann nicht im Katalog.

Das Editieren des Sonderfiles keys wird in Abschnitt 10, Seite 144 beschrieben.

Der Arbeitsfile (EDIT, NAME)

Der Arbeitsfile mit dem Namen *workfile* ist ein temporärer Text- oder BASIC-File, der vom HP-75 erzeugt wird:

- Beim ersten Einsetzen der Batterien und Setzen der Uhr.
- Beim Zurücksetzen des HP-75.
- Nach einem Löschen (PURGE) des momentanen Files.
- Beim Erzeugen eines neuen Text- oder BASIC-Files, dessen Namen nicht im EDIT Befehl spezifiziert wurde.
- Bei der Ausführung eines NAME-Befehls (Seite 64).

Der Arbeitsfile soll Ihnen als «Schmierpapier»-File im Speicher dienen. Sie können einen Arbeitsfile in der gleichen Weise wie jeden anderen Text- oder BASIC-File editieren und bearbeiten. Sie erzeugen einen Arbeitsfile, wenn Sie den Befehl EDIT ohne Angabe eines Filenamens ausführen:

```
:edit text
```

Der Befehl EDIT ohne Filename.

```
workfile    T    0 15:46 02/05/83
```

Erzeugt einen neuen Arbeitsfile des gleichen Typs wie zuvor (Text).

Ein Arbeitsfile kann nur als der *momentane* File existieren; deshalb kann immer nur ein Arbeitsfile im Speicher vorhanden sein. Bevor Sie einen Arbeitsfile verlassen, der zumindest eine Zeile enthält, müssen Sie ihn entweder löschen (mit PURGE) oder benennen.

Wenn Sie bei der Ausführung eines EDIT-Befehls Fehler 69 erhalten – workfile name? – haben Sie versucht, den Filepointer von einem nichtleeren Arbeitsfile wegzubewegen. Geben Sie eine Zeile in den Arbeitsfile ein:

```
3 Der File ist nicht mehr leer.
```

```
:
```

Zeile 3 wurde in den Arbeitsfile eingegeben.

Versuchen Sie jetzt, einen andern File zu editieren, ohne den momentanen File zu benennen.

```
:edit
```

Wenn Sie RTN drücken, wird Fehler 69 – workfile name? – angezeigt. Sie können einen nichtleeren Arbeitsfile nicht verlassen.

Der Befehl NAME bewirkt zweierlei:

- Umbenennen des momentanen Files.
- Erzeugen eines neuen Arbeitsfiles des gleichen Typs, Text oder BASIC.

```
NAME 'Filename'
```

Die Ausführung von NAME hat dieselbe Wirkung wie die Ausführung von RENAME TO 'Filename' und EDIT zusammen. Benennen Sie jetzt den momentanen File und erzeugen Sie einen neuen Arbeitsfile mit dem Befehl NAME:

```
:name 'kopie'
```

Benennen des momentanen Arbeitsfiles.

```
workfile    T    0 15:59 02/05/83
```

Benennt den alten Arbeitsfile in KOPIE um und erzeugt einen neuen Arbeitsfile des selben Filetyps (Text).

Der Befehl NAME ist eine Editierhilfe zur Benennung eines Arbeitsfiles, bevor ein neuer Arbeitsfile erzeugt wird. Sie können NAME jedoch mit jedem File des Speichers ausführen.Sie können zwar einen *leeren* File (Der Katalogeintrag zeigt 0 Bytes) mit NAME benennen, dieser File wird aber aus dem Speicher gelöscht, sobald der Filepointer auf den neuen Arbeitsfile positioniert wird.

Umwandeln von Files (TRANSFORM)

Mit dem Befehl TRANSFORM können Sie Text-Files in BASIC-Files, BASIC-Files in Text-Files, oder beide Filetypen in LIF1-Files (Austauschfiles) umwandeln. Schlagen Sie weitere Information unter «Spezielle Files» in Anhang B, Seite 274 nach.

Zusammenfassung der Befehle

Die folgende Tabelle faßt die 14 Filebefehle zusammen, die in diesem Abschnitt eingeführt worden sind.

Befehl	Parameter*	Operiert auf †	Muß momentaner File sein?	Kann Filepointer bewegen?
AUTO	Ln, Incr	B,T	ja	ja
CAT	Fn**	B,T,A,L, I	nein	nein
COPY	Fn,Fn**	B,T,A,L, I	nein	nein
DELETE	Ln,Ln	B,T	ja	nein
EDIT	Fn	B,T	nein	ja
FETCH	Str,Ln	B,T	ja	ja
FETCH	Ln	B,T	ja	ja
LIST	Fn,Ln,Ln	B,T	nein	nein
MERGE	Fn,Ln,Ln	B,T	ja	nein
NAME	Fn	B,T	ja	ja
PLIST	Fn,Ln,Ln	B,T	nein	nein
PURGE	Fn§	B,T,A,L,I	nein	nein
RENAME	Fn,Fn§	B,T,A,L,I	nein	nein
RENUMBER	Ln,Incr, Ln,Ln	B,T	ja	nein
TRANSFORM	Fn	B,T,I	nein	nein

* Ln = Zeilennummer, 0-9999; Fn = Filename, in einfachen oder doppelten Anführungszeichen; Incr = Inkrementwert, 1-9999; Str = String in Anführungszeichen.

† B = BASIC-File; T = Textfile; A = APPT-File; L = LEX-File; I = Austauschfile.

** Kann einen auf Magnetkarte oder -band abgelegten File spezifizieren.

§ Kann einen auf Magnetband abgelegten File spezifizieren.

Falls der momentane File gelöscht wurde, zeigt der Filepointer auf Zeile 0 eines neuen Arbeitsfile.

Private BASIC-Files, die im Systemkatalog mit PB markiert sind, können nur in den Speicher kopiert, abgearbeitet und gelöscht werden (Seite 117).

Befehlsparameter, die Zeilennummern oder Inkremente spezifizieren, müssen *vorzeichenlose ganze Zahlen* sein. LIST 700,750 ist zum Beispiel ein gültiger Befehl, nicht aber DELETE A,A+50.

Befehlsparameter, die einen String spezifizieren, können beliebige Stringvariablen oder -ausdrücke sein (siehe Abschnitt 5, Seite 78).

Beispiele: Wenn der Stringvariablen B$ der Wert 'DISP' zugewiesen wurde, sucht der Befehl FETCH B$ im momentanen BASIC- oder Text-File nach dem nächsten Auftreten von DISP. Wenn A$ den Wert FINANZ hat, löscht PURGE A$ den File FINANZ.

Stringvariablen dürfen nicht zur Spezifikation von *Schlüsselworten* benutzt werden. Die Worte APPT, KEYS, TEXT und BASIC zum Beispiel müssen explizit in Befehle eingetastet werden.

Hinweis: Filenamen werden gewöhnlich durch das zweite Anführungszeichen abgeschlossen. Das erste *Leerzeichen* nach dem Filenamen kann jedoch zum Abschluß des Filenamens benutzt werden. `edit 'punkte und mittelwerte'` ist zum Beispiel gleichwertig mit `EDIT 'punkte'` und erzeugt den File `PUNKTE`.

Alle oben aufgeführten Befehle sind programmierbar. Siehe dazu Abschnitt 11, Seite 156.

Teil II
Gebrauch des HP-75

Abschnitt 4

Tastenfeld-Berechnungen

Inhalt

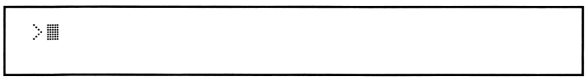
Einführung

Der HP-75 kann arithmetische Ausdrücke (wie 64.32 + 128.16) auf zwei Arten berechnen:

- Direkte Berechnung bei der Eingabe.
- Berechnung des Ergebnisses in einem BASIC-Programm.

In diesem Abschnitt wird die erste Methode, die Benutzung des HP-75 als Tischrechner, behandelt. Der HP-75 rechnet mit zwölfstelliger Genauigkeit im Zahlenbereich von -10^{499} bis $+10^{499}$. *Numerische* (oder algebraische) Variablen und Funktionen werden in Abschnitt 5 diskutiert.

Alle Tastenfeld-Berechnungen werden im EDIT-Modus ausgeführt. Drücken Sie [EDIT]:



Zum Rechnen sollte die BASIC-Eingabeaufforderung in der Anzeige stehen.

Wenn stattdessen die Text-Eingabeaufforderung in der Anzeige erscheint, wird eine Anzeigezeile mit einer Nummer zwischen 0 und 9999 als Textzeile interpretiert und abgespeichert anstatt ausgeführt.*

Alle Tastenfeldoperationen bestehen aus zwei Schritten:

1. Geben Sie die gewünschten Zahlen und Operatoren in die Anzeige ein (ohne Gleichheitszeichen). Bis zu 94 Zeichen können in einem Ausdruck benutzt werden, und Leerzeichen sind ohne Bedeutung.

* Erinnern Sie sich aber, daß Sie auch während der Editierung eines Textfiles arithmetische Operationen durchführen können. Stellen Sie der Zahl in der Anzeige ein Pluszeichen, ein Minuszeichen, einen Dezimalpunkt oder eine linke Klammer voran.

2. Drücken Sie [RTN]. Der Ersetzungs- oder der Einfügungscursor kann sich dabei an beliebiger Stelle in der Anzeige befinden.

Ergebnisse werden mit führendem Leer- oder Minuszeichen, einem Dezimalpunkt (falls erforderlich) und einem nachfolgenden Leerzeichen dargestellt.

Beachten Sie, daß Tastenfeld-Berechnungen keinen Einfluß auf Fileinhalte haben.

Das numerische Tastenfeld des HP-75, auf Seite 28 abgebildet, wird durch Drücken von [CTL] [LOCK] aktiviert. Das numerische Tastenfeld dient zur Beschleunigung von Tastenfeldberechnungen. Das normale Tastenfeld wird durch Drücken von [LOCK] wieder aktiviert.

Arithmetische Operationen

Addition (+)

Verwenden Sie zur Addition zweier oder mehrerer Zahlen die Taste [+].

Beispiel:

```
>2819+3017
```

```
 5836
```

Das Ergebnis wird ohne Eingabeaufforderung und Cursor angezeigt.

Geben Sie keine Kommata ein, da diese nicht als Zifferntrennzeichen, sondern als Zahlentrennzeichen aufgefaßt werden.

Subtraktion (−)

Verwenden Sie zur Subtraktion zweier oder mehrerer Zahlen die Taste [−].

Beispiel:

```
>.051379-1.11750
```

```
-1.066121
```

Die erste Stelle der Anzeige ist für das Vorzeichen reserviert.

Wenn drei oder mehrere Zahlen mit den Operatoren + und − verbunden werden, wird der Ausdruck von links nach rechts bearbeitet.

Beispiel:

```
>24-12-3
```

```
 9
```

Wird als (24–12)–3 berechnet.

Das Minuszeichen wirkt auch als Operator mit *einem* Argument; Sie können es mit einzelnen Zahlen verwenden.

Beispiel:

```
>--3
```

```
 3
```

Gibt die negative Inverse von (–3) zurück.

Multiplikation (*)

Benutzen Sie die Taste [*] zur Multiplikation zweier oder mehrerer Zahlen.

Beispiel:

```
>30.4*12*365
```

```
133152
```

Der Dezimalpunkt wird unterdrückt, da das Ergebnis ganzzahlig ist.

Division (/)

Benutzen Sie zur Division zweier Zahlen die Taste [/].

Beispiel:

```
>4/1.25
```

```
3.2
```

Das Ergebnis. Der HP-75 unterdrückt nachlaufende Nullen.

Wenn drei oder mehr Zahlen mit den Operatoren * und / verknüpft werden, wird der Ausdruck von links nach rechts abgearbeitet.

Beispiel:

```
>24/12*7
```

```
14
```

Zwischenräume sind nicht erforderlich.

Wird als Ausdruck (24/12)*7 berechnet.

Potenzierung (^)

Der Exponentiationsoperator (^) erhebt eine Zahl zur gegebenen Potenz. Um zum Beispiel 5 hoch 3 zu berechnen:

```
>5^3
```

```
125
```

Das Zeichen ^ erhalten Sie mit [SHIFT] [*].

Exponenten können sowohl positiv wie negativ, sowohl ganzzahlig wie gebrochen sein.

Bei Eingabe von Null hoch Null erhalten Sie Warnung 6 –0^0–, und es wird der Ersatzwert 1 zur Verfügung gestellt. Wenn Sie Null zu einer negativen Potenz erheben, erhalten Sie Warnung 5 –0^neg–, und der Ersatzwert $9.99999999999 \times 10^{499}$ (größte Maschinenzahl) wird eingesetzt. Wenn Sie eine negative Zahl zu einer gebrochenen Potenz erheben, erhalten Sie ohne Ersatzwert Fehler 9 –neg^non-integer.

Ganzzahlige Division (DIV und \)

Der Operator DIV teilt zwei Zahlen, gibt aber nur den ganzzahligen Anteil des Quotienten ohne Rundung zurück.

Beispiel:

```
>200 div 3
```

```
66
```

Der Operator DIV.

Größte ganze Zahl, die dreimal in 200 enthalten ist.

Es gilt A DIV B=IP(A/B), wo IP der *ganzzahlige Anteil* von A/B ist.

Sie können die Namen von Operatoren wie DIV und Funktionen in Groß- oder Kleinschreibweise eingeben. Sie können anstelle von DIV auch das Symbol \ verwenden.

Beispiel:

```
>200\3
```

```
 66
```

Der Schrägstrich rückwärts wird mit CTL / erzeugt.

Identisch mit 200 DIV 3.

Siehe auch Seite 83, MOD-Funktion.

Mehrfachrechnungen

Sie können mehrere Ausdrücke gleichzeitig berechnen. Trennen Sie die Ausdrücke mit Kommata oder Semikolons.

Beispiel:

```
>2^7,-.04*100
```

```
 128                    -4
```

Mit dem Komma wird das zweite Ergebnis von Position 22 an dargestellt.

```
>2^7;-.04*100
```

```
 128  -4
```

Mit Semikolon wird das zweite Ergebnis direkt hinter die dem ersten Ergebnis nachfolgende Leerstelle dargestellt.

Letztes Ergebnis (RES)

Dem Ergebnis Ihrer letzten Berechnung wird ein fester Speicherplatz zugeteilt, bis es durch einen neuen berechneten Wert ersetzt wird. Sie können mit der Funktion RES (*result*), die auf diesen Speicherplatz zugreift, den Wert des letzten Ergebnisses jederzeit zurückrufen.

Rufen Sie mit folgender Eingabe das Ergebnis der obenstehenden Rechnung ab:

```
>res
```

```
 -4
```

Die Funktion RES.

Das letzte Ergebnis.

Addieren Sie zu diesem Wert die Zahl 3:

```
>res+3
```

```
 -1
```

RES kann anstelle einer Zahl benutzt werden.

Dieses Ergebnis (–1) wird nun an der RES Speicheradresse abgelegt.

Hinweis: Während einer Programmausführung gibt RES den letzten vom HP-75 angezeigten oder ausgedruckten Wert, nicht notwendigerweise den zuletzt berechneten Wert zurück.

Erinnern Sie sich, daß die Tastenfolge CTL FET den Inhalt des Eingabebuffers in die Anzeige holt. Rufen Sie mit CTL FET den gesamten zu berechneten Ausdruck ab. Sie können diesen Ausdruck dann ergänzen oder editieren und mit Drücken von RTN wieder berechnen.

Arithmetische Hierarchie

Wenn ein Ausdruck zwei oder mehr arithmetische Operationen enthält, werden sie in der folgenden Reihenfolge ausgeführt:

Operator	Funktion	Vorrang
^	Potenzierung.	Zuerst ausgeführt.
*, /, DIV, \	Multiplikation, Division und ganzzahlige Division.	↓
=, – und Negationsminus	Addition, Subtraktion und Negation.	zuletzt ausgeführt.

Zwei oder mehrere Operationen der gleichen Hierarchieebene werden von links nach rechts ausgeführt.

Beispiele:

```
>1+3*2
```

```
7
```

Multiplikation vor Addition.

Berechnet wird 1 + (3*2).

```
>-8+20/4^3
```

```
-7,6875
```

Exponentiation vor Division vor Addition.

Berechnet wird –8 + (20/(4^3)).

```
>24/12/2
```

```
1
```

Division von links nach rechts.

Berechnet wird (24/12)/2.

Das Negations-Minus wird angewandt, sobald ein anderer Operator gleicher oder minderer Rangfolge im Ausdruck erscheint.

Beispiel:

```
>-2*3^2+18
```

```
0
```

Das Negationsminus gilt nur für 2*3^2, da der Operator + von gleicher Rangfolge ist.

Der Ausdruck wird zu –18 + 18.

Klammern ( )

Klammern dienen zu zwei Zwecken:

- Zur Verdeutlichung der Ausführungsreihenfolge mehrdeutiger Ausdrücke. Sie ziehen vielleicht vor, (24/12)/2 einzugeben, um die Divisionsreihenfolge anzudeuten.
- Zur Änderung der normalen Reihenfolge der Auswertung. Die Eingabe von 24/(12/2) zum Beispiel ändert die Reihenfolge der Divisionen.

Klammern besitzen die höchste Priorität in der arithmetischen Hierarchie. Wenn ein Klammernpaar ein anderes enthält, wird das innere Klammernpaar zuerst ausgewertet.

Beispiel:

```
>1.8+(5*(4-2))^3
```

```
1001.8
```

Enthält zwei verschachtelte Klammerpaare.

4–2 wird zuerst berechnet, dann das Produkt mit 5.

Der HP-75 entfernt überflüssige Klammern in Programmanweisungen automatisch.

Klammern allein werden nicht als Multiplikationsanweisung verstanden. 3(9–5) zum Beispiel muß als `3*(9-5)` eingegeben werden. Die eckigen Klammern ([und]) werden in Tastenfeldberechnungen überhaupt nicht benutzt.

Zahlengenauigkeit

Wenn Sie numerische Daten in den HP-75 eingeben, sei es über das Tastenfeld oder mit einer `DATA`-Anweisung (siehe Abschnitt 14) innerhalb eines Programms, liest der Computer 13 signifikante Stellen der Eingabe, rundet diese auf 12 Stellen und speichert das Ergebnis.

Die folgenden Beispiele dienen zur Erläuterung der Datengenauigkeit:

Geben Sie ein:

```
>12.3456789012345
```

Nach [RTN] gespeicherter Wert:

```
12.3456789012
```

Die 13. Stelle ist kleiner als 5, die 12. Stelle wird also abgerundet. Beachten Sie, daß die 14. Stelle beim Runden keine Rolle spielt.

```
>12.34567890125 6
```

```
12.3456789013
```

Die 13. Stelle ist größer gleich 5, deshalb wird die zwölfte Stelle aufgerundet.

Der HP-75 rechnet intern mit 15 signifikanten Stellen, aber die Ergebnisse werden auf 12 signifikante Stellen gerundet gespeichert und angezeigt. Bei der Rundung wird die 12. Stelle nur dann aufgerundet, wenn die 13. Stelle größer oder gleich 5 ist.

Zahlenformate

Der HP-75 verwendet Zahlenformate entsprechend der Norm des American National Standards Institute.

Gleitkommaformat

Zahlen, die mit 12 oder weniger Stellen vollständig dargestellt werden können, werden im *Gleitkommaformat* angezeigt. Der Dezimalpunkt kann an beliebiger Stelle innerhalb der Zahl auftreten. Nachlaufende Nullen rechts des Dezimalpunkts werden unterdrückt.

Beispiel:

```
>32.10000*2
```

```
64.2
```

Nachlaufende Nullen werden unterdrückt.

Führende Nullen werden ebenfalls unterdrückt.

Beispiel:

```
>040350.3 + 10
```

```
40360.3
```

Die führende Null wird unterdrückt.

Alle signifikanten Stellen einer Zahl, bis zu 12 Stellen, werden angezeigt.

Beispiel:

```
>.123456789012345
```

```
.123456789012
```

Eine Zahl mit 15 Stellen.

Auf 12 Stellen gerundet.

Zahlen mit Absolutbetrag zwischen 1 und 1 Billion (10^{12}) werden mit bis zu 12 signifikanten Stellen und ohne Exponenten angezeigt.

Beispiel:

```
>-999988887777.6666
```

```
-999988887778
```

Eingabe einer 16stelligen Zahl.

Die 12 führenden Stellen werden angezeigt.

Zahlen zwischen –1 und + 1 werden vollständig und ohne Exponenten ausgegeben, wenn sie mit 12 oder weniger Stellen exakt dargestellt werden können.

Beispiel:

```
>0.00112233
```

```
.00112233
```

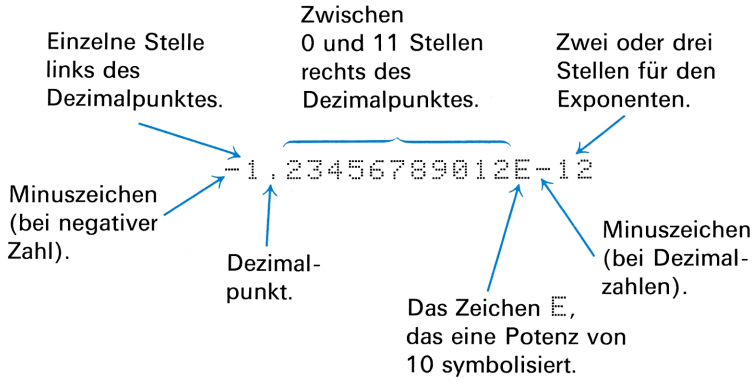
Eine Dezimalzahl mit 8 gültigen Stellen.

Alle gültigen Stellen werden angezeigt.

Alle anderen Zahlen werden in Exponentialdarstellung angezeigt.

Exponentialdarstellung (E)

Die Exponentialdarstellung, auch als wissenschaftliche Notation bezeichnet, ist eine abgekürzte Schreibweise für Zahlen, die zu groß oder zu klein sind, um normalerweise in die Anzeige zu passen, d.h. Zahlen, die mit 12 Stellen nicht vollständig dargestellt werden können. Die Zahl -.00000000000123456789012 hat beispielsweise die Exponentialdarstellung:



Exponentialdarstellungen bestehen aus zwei Teilen, der *Mantisse*, die die signifikanten Stellen enthält, und dem *Exponenten* zur Basis 10.

Beispiele:

```
>1000000000000000000000000
```

Eine 1 mit 24 Nullen.

```
1.E24
```

Als Potenz von 10 mit einer signifikanten Stelle ausgedrückt.

```
>.000000000000547
```

Eine kleine positive Zahl.

```
5.47E-13
```

Als negative Potenz von 10 mit drei signifikanten Stellen ausgedrückt.

Sie können auch Zahlen in Exponentialdarstellung *eingeben*. Der HP-75 gibt jedoch nur dann eine Zahl in Exponentialdarstellung aus, wenn die Gleitkommadarstellung nicht ausreicht.

Beispiele:

```
>28.19e2
```

```
2819
```

Positive Mantisse und Exponent.

```
>2819e-3
```

```
2.819
```

Positive Mantisse, negativer Exponent.

```
>-28.19e16
```

```
-2.819E17
```

Zahl überschreitet 12 Stellen.

Exponenten müssen ganze Zahlen zwischen –499 und + 499 sein. Das Pluszeichen für positive Mantissen und Exponenten ist optional.

Sie können auch Ausdrücke eingeben, die sowohl Zahlen in Gleitkomma als auch in Exponentialdarstellung enthalten.

Beispiele:

```
>1e12-1000
```

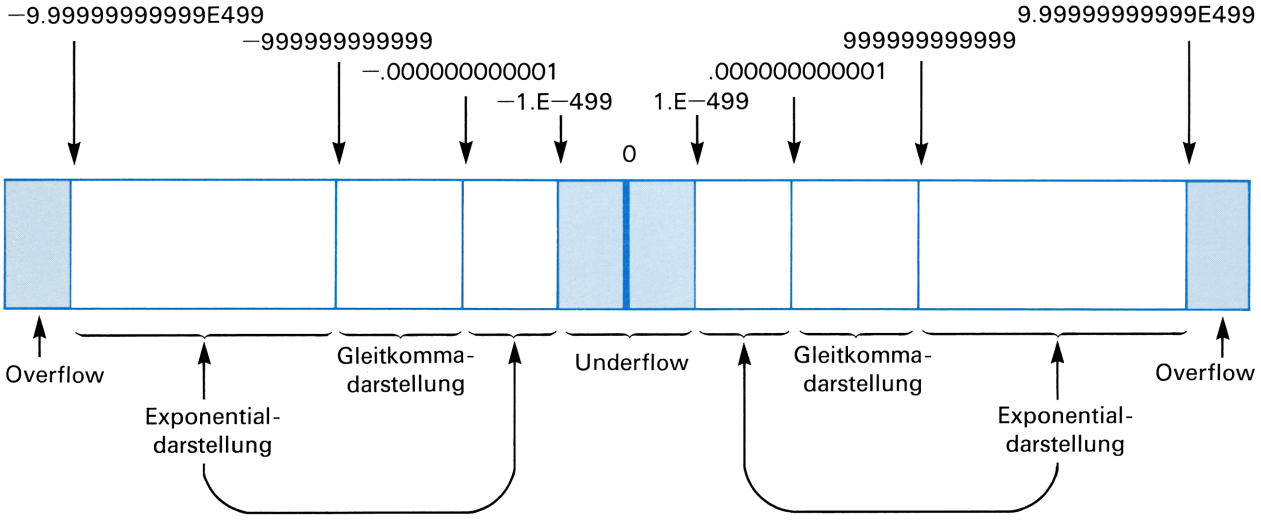
```
999999999000
```

```
>1e12+1000
```

```
1.000000001E12
```

Zahlenbereich (INF, EPS)

Das folgende Diagramm zeigt den Bereich der Zahlen, die Sie eingeben und speichern können:



Die größte Zahl des HP-75 ist $9.99999999999 \times 10^{499}$. Diesen Wert können Sie mit der Funktion INF (*infinity*) erzeugen:

```
>inf
```

```
9.99999999999E499
```

Erzeugt die größte Maschinenzahl.

Die größte negative Zahl ist der Negativwert von INF:

```
>-inf
```

```
-9.99999999999E499
```

Die kleinste positive Zahl des Computers ist $1x10^{-499}$. Diesen Wert können Sie mit der Funktion EPS (*epsilon*) erzeugen:

```
>eps
```

```
1.E-499
```

Gibt die der Null am nächsten liegende positive Zahl zurück.

Die kleinste negative Zahl schließlich ist der Negativwert von EPS:

```
>-eps
```

```
-1.E-499
```

Wenn Sie einen Wert größer als INF oder kleiner als −INF benutzen, erhalten Sie die Warnung 2 –num too large– und einen Ersatzwert von plus/minus maschinenunendlich. Ein von Null verschiedener Wert zwischen EPS und −EPS erzeugt Warnung 1 –num too small– und den Ersatzwert 0. (siehe auch Seite 89, DEFAULT ON und DEFAULT OFF.)

Abschnitt 5

Numerische Funktionen und Ausdrücke

Inhalt

Einführung

In diesem Kapitel werden die *numerischen*, oder mathematischen, Funktionen des HP-75 diskutiert, die über einfache Arithmetik hinausgehen. Alle numerischen Berechnungen werden im EDIT-Modus ausgeführt, entweder über das Tastenfeld oder während einer Programmausführung. In den folgenden Beispielen wird die BASIC- Eingabeaufforderung benutzt; Sie können jedoch auch während der Editierung eines Textfiles numerische Berechnungen durchführen.

Variablen

Häufig ist es sinnvoll, eine Zahl oder einen Zeichenstring durch einen *Namen* anstelle eines bestimmten Werts darzustellen. Eine solche Darstellung ist auf dem HP-75 mit Hilfe von Variablen möglich. Eine Variable ist einfach ein Name für einen Speicherplatz, der numerische oder Stringdaten enthält. Variablen sind für die Programmierung von grundlegender Bedeutung.

Sie können mit dem HP-75 Werte auf Variablen zuweisen und Werte von Variablen berechnen, ändern, vergleichen und ausgeben.

Der HP-75 unterscheidet drei Variablentypen:

- einfache numerische Variablen, wie zum Beispiel X und X1.
- numerische Feldvariablen, wie zum Beispiel A() und A1(,). Felder können gleichartige Daten aufnehmen (zum Beispiel eine Reihe von Temperaturmessungen) und dienen zur bequemen Handhabung von Listen und Tabellen numerischer Daten in Programmen. Siehe auch Abschnitt 13.

- Stringvariablen, wie zum Beispiel S$ und S1$. Stringvariablen speichern alphanumerische Information (zum Beispiel 'Monatsgehalt:') beliebiger Länge, von null Zeichen bis zu einem Maximum, das nur durch den verfügbaren Speicherplatz begrenzt ist. Stringvariablen können Filenamen, HP-IL-Einheitscodes, Tastendefinitionen usw. spezifizieren sowie zur Eingabe, Verarbeitung und Ausgabe von alphanumerischer Information verwendet werden.

Der Einfachheit halber beschränken sich die Beispiele auf Tastenfeld- oder Rechnervariablen; Programmvariablen unterliegen analogen Regeln.

Benennen einfacher numerischer Variablen

Als Namen für einfache numerische Variablen sind zulässig:

- Ein beliebiger Buchstabe von A bis Z. (In den Beispielen werden Kleinbuchstaben benutzt; diese werden aber vom System als Großbuchstaben interpretiert.)
- Ein beliebiger Buchstabe, direkt gefolgt von einer Ziffer zwischen 0 und 9.

Zulässige Namen sind zum Beispiel a1, C, z, J5, C3 und p0. Insgesamt können 286 einfache numerische Variablen benannt werden.

Wertzuweisung auf Variablen ([=], LET)

Mit den Tasten [=] und [RTN] können Sie Werte auf Variablen zuweisen.

Beispiele:

```
>a=15
```

Durch Drücken von [RTN] wird der Wert 15 in einer Variablen (einem Speicherplatz) mit dem Namen A gespeichert.

```
>x3=2*25
```

Der Variablen X3 wird der *rechts* des Gleichheitszeichens stehende Wert 50 zugewiesen.

Zur Wertzuweisung auf Variablen können Sie auch den Befehl LET verwenden.

Beispiel:

```
>let b6=x3
```

Durch Drücken von [RTN] wird B6 der momentane Wert der Variablen X3 zugewiesen.

Beachten Sie, daß das Schlüsselwort LET optional ist; a=15 ist gleichwertig mit let a=15.

Wenn den Variablen Werte zugewiesen wurden, können diese anstelle von Zahlen in Tastenfeld- und Programmberechnungen benutzt werden.

Beispiele:

```
>a^2
```

Potenzierung (^) einer Variablen.

```
225
```

Das Ergebnis von 15^2.

```
>x3/b6
```

Verknüpfung der Werte zweier Variablen mit dem Divisionsoperator.

```
1
```

Das Ergebnis von 50 geteilt durch 50.

Variablen können sehr einfach neue Werte zugewiesen werden. Um zum Beispiel den Wert von A auf 16 zu ändern, können Sie a=a+1, a=16 oder eine ähnliche Zuweisung eingeben. Sie können den Wert einer Variablen anzeigen, indem Sie ihren Namen eintasten und RTN drücken.

Beispiel:

```
>x3
```

Anzeige des Werts einer Variablen.

```
50
```

Der momentane Wert von x3.

Sie können die Werte mehrerer Variablen in der gleichen Zeile anzeigen, wenn Sie die Variablen mit Kommata oder Semikolons trennen.

Beispiel:

```
>a;x3,b6
```

Verwenden Sie Semikolons für enge Zwischenräume und Kommata für 22-Spaltenabstände.

```
15  50          50
```

Anzeige der drei Variablenwerte.

Sie können auch mehreren Variablen in einer Zeile einen Wert zuweisen. Trennen Sie die Variablennamen durch *Kommata*.

Beispiel:

```
>a,b,c=0
```

Beim Drücken von RTN nehmen die Variablen A, B und C den Wert 0 an.

Rufen Sie die einzelnen Variablen ab, um die Zuweisung zu prüfen:

```
>a;b;c
```

```
0 0 0
```

Die Variablen haben den gleichen Wert.

Genauigkeit numerischer Variablen (REAL, SHORT, INTEGER)

Neben dem Namen und dem Wert einer Variablen können Sie auch ihre Genauigkeit, das heißt, die Anzahl von Stellen, mit der der HP-75 ihren Wert darstellt, bestimmen. Drei Genauigkeitsdeklarationen stehen zur Verfügung: REAL, SHORT und INTEGER.

- REAL Variablenwerte werden mit der vollen Genauigkeit des HP-75 gespeichert. Sie umfassen den gesamten Wertebereich von $-9.99999999999 \times 10^{499}$ bis $-1.00000000000 \times 10^{-499}$, 0 und $1.00000000000 \times 10^{-499}$ bis $9.99999999999 \times 10^{499}$. REAL Zahlen werden intern mit einer zwölfstelligen Mantisse und einen dreistelligen Exponenten dargestellt.
- SHORT Variablenwerte umfassen einen engeren Bereich, von -9.9999×10^{99} bis -1.0000×10^{-99}, 0 und von 1.0000×10^{-99} bis 9.9999×10^{99}. SHORT Zahlen werden dementsprechend intern mit fünf Stellen und zweistelligem Exponenten dargestellt.
- INTEGER Werte liegen zwischen –99999 und +99999. INTEGER Zahlen werden fünfstellig ohne Exponenten gespeichert.

Deklarieren Sie Variablen mit REAL, SHORT oder INTEGER, um deren Genauigkeit zu spezifizieren.

Beispiele:

```
>real p
```

Deklariert P als REAL Variable.

```
>short x,y,z
```

Deklariert X, Y und Z als SHORT Variablen.

```
>integer i,j,k
```

Deklariert I, J und K als INTEGER Variablen.

Weisen Sie jeder dieser Variablen den Wert π zu, um die Deklaration zu prüfen:

```
>p,x,i=pi
```

Weist den Variablen den Wert π entsprechend ihrer Genauigkeit zu.

Rufen Sie danach die einzelnen Variablenwerte ab:

```
>p,x,i
```

```
3.14159265359 3.1416 3
```

Die Werte werden entsprechend der Genauigkeitsdeklaration angezeigt.

Beachten Sie, daß ein Wert gerundet wird, wenn die Anzahl der Ziffern die Genauigkeit der Variablen überschreitet.

Allen numerischen Variablen wird volle Genauigkeit (REAL) zugewiesen, wenn Sie nicht implizit anderes deklarieren. Die Werte der Variablen A, X3, B6, B und C, die zuvor auftraten, wurden alle mit REAL Genauigkeit abgespeichert, obwohl Sie deren Genauigkeit bei der Wertzuweisung nicht spezifizierten.

Auf dem HP-75 können Sie die Genauigkeit einer Variablen nicht mehr ändern, nachdem Sie diese deklariert haben oder der Variablen einen Wert zugewiesen haben. Wenn Sie jetzt zum Beispiel `short p` eingeben, erhalten Sie die Fehlermeldung 35 -DIM exist var- die Sie darauf hinweist, daß Sie versucht haben, die Genauigkeit einer existierenden Variablen nachträglich zu deklarieren.

REAL Variablen bieten Ihnen zwar größte Genauigkeit und den weitesten Rechenbereich; Sie sparen aber Speicherplatz, wenn Sie SHORT oder INTEGER Variablen deklarieren. Siehe dazu auch Anhang D, «Systemspeicheranforderungen».

Rechner- und Programmvariablen (CLEAR VARS)

Die Werte von Rechnervariablen – die vom Tastenfeld aus zugewiesenen Werte – bleiben im Speicher erhalten, bis Sie sie ändern oder löschen. Sie können also Rechnervariablen für den Dauergebrauch speichern. Sie können zum Beispiel zuerst der Variablen A die Loschmidtsche Zahl über das Tastenfeld zuweisen, dann ein Programm abarbeiten, und danach den Wert A wieder abrufen und in weiteren Tastenfeld-Berechnungen benutzen.

Beachten Sie, daß ein Unterschied zwischen Programm- und Rechnervariablen besteht. Den Programmvariablen werden nur während der Programmausführung Werte zugewiesen. Ein Programm benutzt Variablen streng *lokal* im Programm selbst und kann nicht auf Rechnervariablen zugreifen. Wenn die Programmausführung aus irgendeinem Grund, zum Beispiel nach einem Drücken der Taste [ATTN] oder nach Ausführung einer STOP-Anweisung, unterbrochen ist, sind Programmvariablen direkt über das Tastenfeld zugreifbar. Wenn zum Beispiel eine Rechnervariable A und eine Programmvariable A existieren, können Sie nach einer Unterbrechung des Programms den Wert der *Programm*variablen und nicht der Rechnervariablen abfragen. Sie *können* jedoch in diesem Fall auf Rechnervariablen zugreifen, die bei Programmvariablen nicht auftretende Namen besitzen. Nach Beendigung des Programms können Sie wieder alle vorhergehenden Rechnervariablen abrufen. Siehe auch Abschnitt 11, «Programmvariablen».

Mit dem Befehl CLEAR VARS können Sie alle Rechner- und Programmvariablen löschen; der von ihnen belegte Speicherplatz steht dann wieder zur Verfügung.

```
CLEAR VARS
```

Den Befehl CLEAR VARS verwenden Sie wahrscheinlich häufiger (zum Beispiel jetzt), damit die Variablen nicht unnötig viel Speicherplatz belegen. (CLEAR VARS beinhaltet auch eine «Deallokation» initialisierter Programme, siehe Seite 158.) Wenn Sie sich auf eine gewisse Anzahl von *Namen* für Rechnervariablen beschränken, können Sie den von den Variablen belegten Speicherplatz klein halten.

Numerische Funktionen

Numerische Funktionen sind fest eingebaute Routinen, die aus numerischen oder Stringwerten einfache Zahlen berechnen. Der HP-75 ist mit mehr als 40 vordefinierten numerischen Funktionen ausgerüstet; sie alle können in Programmen und auch über das Tastenfeld ausgeführt werden. Einige davon, wie zum Beispiel NUM und MEM wurden zuvor schon besprochen. Eine vollständige Liste finden Sie in Anhang H.

Die Information, die zur Ausführung einer Funktion benötigt wird, bezeichnet man als *Argument* der Funktion; die Funktionen des HP-75 können auf ein oder mehrere, oder auch auf kein Argument wirken. Ein Argument kann selbst eine Variable, eine andere Funktion oder ein ganzer Ausdruck sein, sofern deren Berechnung einen konstanten Wert ergibt.

HP-75 Funktionen werden wie folgt über das Tastenfeld ausgeführt:

1. Geben Sie den Namen der Funktion mit Groß- oder Kleinbuchstaben ein.
2. Schließen Sie das Argument, falls erforderlich, in Klammern ein. Wenn die Funktion mehrere Argumente benötigt, trennen Sie diese mit Kommata.
3. Drücken Sie [RTN], um das Ergebnis zu berechnen.

In den folgenden Unterabschnitten werden die numerischen Funktionen des HP-75 entsprechend ihrer Anwendung unterteilt.

Funktionen zur Zahlenmanipulation (ABS, IP, FP, INT, FLOOR, CEIL)

Die folgende Tabelle enthält den Namen, das Argument und die Bedeutung von sechs Funktionen, die Zahlen abändern. Jede Funktion wird zusammen mit dem Wert aufgelistet, den ein einfaches Argument X ergibt, wobei X eine Konstante (wie 112.75), eine Variable (wie A), eine andere Funktion (wie SQR(A)) oder ein Ausdruck (wie 112.75*SQR(A)) sein kann.

Funktion und Argument	Bedeutung	Beispiel
ABS(X)	Absolutbetrag von X.	abs(-235) 235
IP(X)	Ganzzahliger Anteil von X; der Teil links des Dezimalpunktes.	ip(10/3) 3
FP(X)	Gebrochener Anteil von X; der Teil rechts des Dezimalpunktes.	fp(10/3) .3333333333
INT(X)	Größte ganze Zahl kleiner oder gleich X.	int(-7.23) -8
FLOOR(X)	Gleiche Funktion wie INT(X).	floor(7.23) 7
CEIL(X)	Kleinste ganze Zahl größer oder gleich X.	ceil(7.23) 8

Beachten Sie den Unterschied zwischen den Funktionen IP, FLOOR (oder INT) und CEIL. Bei einem positiven Argument sind die Ergebnisse von IP und FLOOR identisch; bei einem negativen Argument sind die Werte von IP und CEIL identisch.

Allgemeine Funktionen (SQR, MOD, SGN, MAX, MIN, RMD, PI, INF, EPS, RND)

Vier der folgenden zehn Funktionen wirken auf zwei Argumente (durch Kommata getrennt), und vier benötigen gar kein Argument. Außer RND sind die argumentlosen Funktionen *konstante* Funktionen, da sie bei jeder Ausführung den gleichen Wert ergeben.

Funktion und Argument	Bedeutung	Beispiel
SQR(X)	Positive Quadratwurzel von X.	sqr(16.1) 4.01248052955
MOD(X,Y)	Ganzzahliger Rest von X/Y; d.h. X-Y*INT(X/Y).	mod(10,3) 1
SGN(X)	Vorzeichen von X: 1 bei positivem X, -1 bei negativem X, 0 bei X=0	sgn(-5) -1
MAX(X,Y)	Maximum zweier Werte; der größere Wert wird zurückgegeben	max(4.5,4.67) 4.67
MIN(X,Y)	Minimum zweier Werte; der kleinere Wert wird zurückgegeben	min(-3,-2.99) -3
RMD(X,Y)	Rest bei X/Y; d.h. X-Y*IP(X/Y)	rmd(10,3) 1
PI kein Argument	Zwölfstellige Näherung von Pi.	pi 3.14159265359
INF kein Argument	Größte Maschinenzahl des HP-75.	inf 9.99999999999E499
EPS kein Argument	Epsilon – die kleinste positive Zahl des HP-75.	eps 1.E-499
RND kein Argument	Zufallszahl – erzeugt die nächste Zahl *R* in einer Folge von Pseudo-Zufallszahlen mit $0 \leqslant R < 1$.	rnd .011750051479

Wenn X und Y beide positiv sind, wirken RMD(X,Y) und MOD(X,Y) gleich. Die Ergebnisse können bei verschiedenem Vorzeichen jedoch unterschiedlich sein. Zum Beispiel ist RMD(-37,7) gleich -2, während MOD(-37,7) gleich 5 ist.

Die Funktion RND erzeugt bei jeder Ausführung eine neue Pseudozufallszahl; dieser Wert ist größer oder gleich 0 und kleiner als 1.

Der *Startwert* einer Zufallszahlenfolge bestimmt die Werte, die RND erzeugt. Bei jeder Ausführung kombiniert RND die letzte Zufallszahl mit einem vorgegebenen Multiplikator und erzeugt so eine neue Zufallszahl. Der Startwert kann jederzeit mit der Anweisung RANDOMIZE über das Tastenfeld oder in einem Programm vorgegeben werden.

RANDOMIZE [*numerischer Ausdruck*]

Sie können den Startwert des Zufallszahlengenerators wie folgt spezifizieren:

- Führen Sie RANDOMIZE ohne Argument aus; der Startwert wird dann mit Hilfe der Systemuhr erzeugt.
- Geben Sie eine Konstante oder einen Ausdruck im Zahlenbereich des HP-75 vor, wenn Sie die Zufallszahlenfolge auf einem bestimmten Wert basieren lassen wollen.

Benutzen Sie zum Beispiel 423 als Startwert einer Zufallszahlenfolge:

```
>randomize 423
```

```
>
```

Die Funktion RND erzeugt nun eine neue Folge von Zufallszahlen.

Erzeugen Sie die erste Zufallszahl dieser Folge:

```
>rnd
```

```
.629385058782
```

Erste Zahl der RND-Folge, auf dem Startwert 423 basierend.

Mit dem gleichen Startwert erzeugen Sie in Ihren Programmen immer die identische Folge von Zufallszahlen. (Der Startwert Null erzeugt eine konstante Nullfolge.)

Bei fehlender RANDOMIZE-Anweisung stellt der HP-75 der Funktion RND automatisch bei jeder Programmausführung* einen Ersatzstartwert (.529199358633) zur Verfügung.

Das Programm SINGSONG in Abschnitt 1 zum Beispiel erzeugt ohne RANDOMIZE-Anweisung immer die gleiche Tonfolge. Der Ersatzwert für den Startwert wird auch vorgegeben, wenn der Befehl RND nach der Ausführung eines Programms, das den Zufallszahlengenerator nicht benutzt, über das Tastenfeld eingegeben wird.

Mit der folgenden Formel können Sie ganzzahlige Zufallszahlen i_1, i_2... i_j... mit $A \leqslant i_j \leqslant B$ erzeugen, wobei A und B beliebige reelle Zahlen (mit $|A|, |B| < 10^{13}$) darstellen:

$$i_j = \text{IP}((B + 1 - A) * \text{RND} + A)$$

Erzeugen Sie beispielsweise eine Zufallszahl zwischen 1 und 100:

```
>ip(100*rnd+1)
```

Gesucht ist die nächste Zufallszahl zwischen 1 und 100.

```
75
```

Das Ergebnis, basierend auf der zweiten Zufallszahl der obenstehenden Folge.

Bei statistisch signifikanten Stichprobengrößen können gute statistische Eigenschaften** des Zufallszahlengenerators erwartet werden.

Logarithmische Funktionen (LOG, EXP, LOG10)

Der HP-75 berechnet natürliche und Zehnerlogarithmen und deren Umkehrungen.

Funktion und Argument	Bedeutung	Beispiel
LOG(X)	Ln X – natürlicher Logarithmus eines positiven Werts X zur Basis *e*.	log(26) 3.25809653802
EXP(X)	e^X, natürlicher Antilogarithmus.	exp(1) 2.71828182846
LOG10(X)	lg(X) – gewöhnlicher Logarithmus eines positiven Werts X zur Basis 10.	log10(1000) 3

* Beim Aufruf eines anderen Programms durch ein Programm wird die im ersten Programm begonnene Zufallszahlenfolge im zweiten Programm fortgesetzt, falls das zweite Programm nicht eine eigene RANDOMIZE Anweisung (Seite 234) ausführt.

** Der Zufallszahlengenerator des HP-75 genügt dem Spektraltest. (Donald E. Knuth, *The Art of Computer Programming* (Massachusetts, 1969), vol.2, section 3.4.)

Der Antilogarithmus zur Basis 10 (10^x) kann mit dem Exponentialoperator `10^X` berechnet werden.

Trigonometrische Befehle (OPTION ANGLE RADIANS, OPTION ANGLE DEGREES)

Setzen Sie den HP-75 mit einem der beiden folgenden Befehle auf den gewünschten trigonometrischen Modus, bevor Sie trigonometrische Funktionen ausführen.

```
OPTION ANGLE RADIANS
```

Ein Vollkreis umfaßt 2 π Radiant. Dies ist die Voreinstellung des HP-75 nach einem Zurücksetzen des Systems.

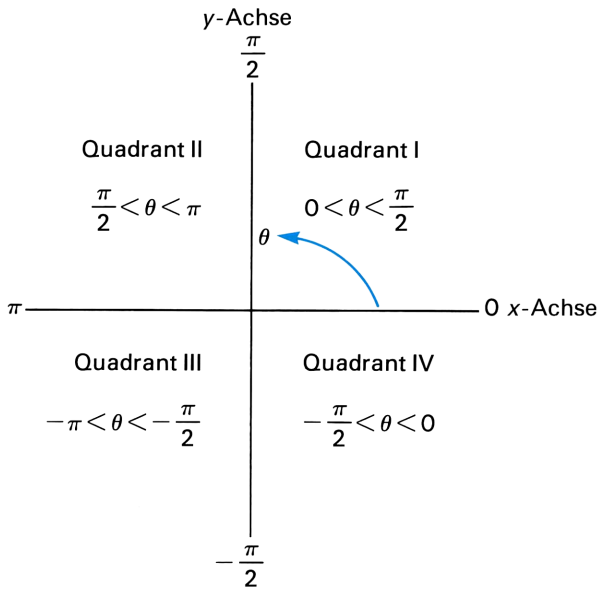
```
OPTION ANGLE DEGREES
```

Ein Vollkreis umfaßt 360 (Alt-)Grad.

Die derzeitige Einstellung des trigonometrischen Modus bleibt solange erhalten, bis Sie sie mit einem anderen `OPTION ANGLE` Befehl ändern.

Trigonometrische Funktionen (SIN, ASIN, COS, ACOS, TAN, ATAN, ANGLE, CSC, SEC, COT, RAD, DEG)

Der HP-75 verfügt über 12 vordefinierte trigonometrische Funktionen. Sie sollten sich den Wertebereich der *Umkehrfunktionen* (arcsin, arccos, arctan), der alle vier Quadranten umfassen kann, vor Augen führen. Im *Radiant*-Modus werden Winkel vom HP-75 wie folgt dargestellt:

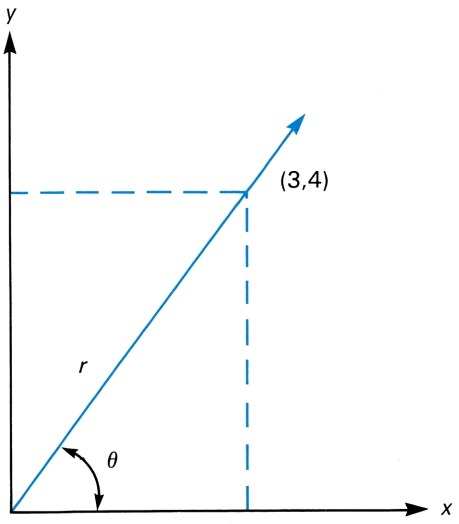


Funktion und Argument	Bedeutung	Beispiel (bei Einstellung OPTION ANGLE RADIANS)
SIN(X)	Sinus von X	sin(pi/2) 1
ASIN(X)	Arcussinus von X, mit $-1 \leqslant X \leqslant 1$. In Quadrant I oder IV.	asin(1) 1.57079632679
COS(X)	Cosinus von X.	cos(0) 1
ACOS(X)	Arcuscosinus von X, wobei $-1 \leqslant X \leqslant 1$. In Quadrant I oder II.	acos(1) 0
TAN(X)	Tangens von X.	tan(pi/4) 1
ATN(X)	Arcustangens von X. In Quadrat I oder IV.	atn(1) .785398163397
ANGLE (X,Y)	Arctan (X/Y), im richtigen Quadranten; d.h. der Winkel θ zwischen (X,Y) und der X-Achse.	angle(sqr(3),1) .523598775598
CSC(X)	Cosecans von X.	csc(pi/6) 2
SEC(X)	Secans von X	sec(pi/3) 2.00000000001
COT(X)	Cotangens von X.	cot(pi/6) 1.73205080757
RAD(X)	Umwandlung von Grad in Radiant.	rad(45) .785398163397
DEG(X)	Umwandlung von Radiant in Grad.	deg(pi) 180

Die Funktionen ATN und ANGLE dienen zu Transformationen zwischen kartesischen und Polarkoordinaten. Bei gegebenen Koordinaten x und y eines Punktes in Quadrant I oder IV ergeben sowohl ATN als auch ANGLE den Wert des Winkels zwischen der x- Achse und der Ursprungsgeraden durch den Punkt.

Beispiel: Wandeln Sie die kartesischen Koordinaten (3,4) in Polarkoordinaten um, und geben Sie den Winkel in Grad an.

Quadrant I



Lösung:

Nach dem Satz von Pythagoras gilt:

$r = \sqrt{x^2 + y^2}$

Lösen Sie mit ATN oder ANGLE nach θ.

θ=ATN(4/3) oder θ=ANGLE(3,4)

Mit diesen Schritten erhalten Sie die Lösung:

```
option angle degrees
```

Mit RTN setzen Sie den HP-75 auf Grad-Modus.

```
>sqr (3^2+4^2)
```

```
5
```

Lösen nach *r*.

```
>angle(3,4)
```

```
53.1301023542
```

Lösen nach θ.

Der Wert von θ in dezimalen Grad.

Beachten Sie den Unterschied zwischen den Funktionen ANGLE und ATN. ANGLE benötigt zwei Argumente, um den richtigen Quadranten des Arcustangens zu finden. ATN gibt den Hauptwert, das heißt, den Wert im ersten oder vierten Quadranten, des Arcustangens eines einzigen Arguments zurück. ANGLE(-3,-2) gibt zum Beispiel –146.309932474 Grad (in Quadrant III) zurück, während ATN(-2,-3) den Wert 33.690067526 Grad (im ersten Quadranten) findet.

Numerische Ausdrücke

Numerische Ausdrücke erscheinen in jeder mathematischen Operation. Die einfachsten numerischen Ausdrücke sind Konstanten, wie 5. Einfache numerische Variablen (wie X3) und Funktionen (wie SQR(2)) stellen ebenfalls einfache Ausdrücke dar. Ein numerischer Ausdruck kann jedoch aus einer beliebigen Anzahl von Konstanten, Variablen und Funktionen bestehen, die durch Operatoren und Klammern verknüpft sind. In Abschnitt 4 wurden arithmetische Operatoren wie + und ^ besprochen, die Ausdrücke verbinden. Die folgenden Abschnitte behandeln zwei andere Arten von Operatoren – Vergleichs- und logische Operatoren – die Ausdrücke in Relation zueinander setzen.

Vergleichsoperatoren (=, <>, #, >, >=, <, <=)

Die sieben Vergleichsoperatoren des HP-75 vergleichen die Werte zweier Ausdrücke und geben eine 1 zurück, wenn der Vergleich wahr ist, eine 0, wenn er falsch ist. Das bedeutet, die Vergleichsoperatoren wirken auf numerischen Werten und geben Boole'sche Werte zurück.

Vergleichs-Operator	Vergleich	Beispiel
=	Gleich?	1=1 1
<> und #	Ungleich?	3<>4 1
>	Größer als?	1>-1 1
>=	Größer oder gleich?	3.14>=pi 0
<	Kleiner?	3.14<pi 1
<=	Kleiner oder gleich?	-eps<=0 1

Beachten Sie, daß das Gleichheitszeichen sowohl bei Wertzuweisungen auf Variablen wie auch in Vergleichsanweisungen verwendet wird. Wann immer eine Eingabe auf beide Arten interpretiert werden kann, wird sie vom HP-75 als Wertzuweisung aufgefaßt.

Beispiele:

```
a=3
```

Wird A gleich 3 gesetzt oder mit 3 verglichen?

```
>
```

Wird als Wertzuweisung aufgefaßt.

```
>(a=3)
```

Klammern spezifizieren einen numerischen Ausdruck.

```
 1
```

Beziehung ist wahr.

```
>3=a
```

Umgekehrte Reihenfolge erzwingt die Behandlung als Vergleich.

```
 1
```

Zeigt, daß A derzeit den Wert 3 hat.

```
>b=a=3
```

Mit RTN wird B der Wert (a=3), d.h. 1, zugewiesen.

Da ein Gleichheitszeichen sowohl eine Wertzuweisung als auch eine Vergleichsoperation bedeuten kann, sollten Ausdrücke mit Gleichheitszeichen eindeutig sein. Vergleichsoperatoren werden häufig zur Kontrolle der Reihenfolge einer Programmausführung (Abschnitt 12, Seite 177) benutzt und können auch zum Vergleich von Stringvariablen verwendet werden (Abschnitt 13, Seite 203).

Die Vergleichsoperatoren bewerten die Beziehung mit Hilfe einer Subtraktion. Zahlenwerte, die bei Subtraktion eine Fehlerbedingung erzeugen würden, erzeugen den gleichen Fehler bei allen Vergleichsoperationen.

Logische Operatoren (AND, OR, EXOR, NOT)

Die vier logischen Operatoren operieren mit Boole'schen Werten und geben Boole'sche Werte zurück. Die logischen Operatoren interpretieren alle von 0 verschiedenen Operanden als 1 oder wahr, und alle mit Null identischen Operanden als 0 oder falsch. AND, OR und EXOR erzeugen den Wert 1, wenn das Verhältnis zwischen den Operanden wahr ist, sonst 0. NOT, der Negationsoperator, gibt den umgekehrten Wert (0 oder 1) eines einzelnen Operanden zurück.

Logischer Operator	Bewertungsgrundlage	Beispiel
AND	Beide Ausdrücke wahr (ungleich Null)?	`3 and 4` ` 1`
OR	Einer der Ausdrücke wahr?	`3 or 0` ` 1`
EXOR	Einer und nur einer der beiden Ausdrücke wahr? Äquivalent zu (A AND NOT B) OR (B AND NOT B).	`3 exor 0` ` 1`
NOT	Ist der Ausdruck falsch (d.h. 0)?	`not 1` ` 0`

Mit Vergleichs- und logischen Operatoren können Sie numerische Konstanten (3 OR 0), Variablen (A AND B), Funktionen (SIN(A) AND COS(A)) und sogar kombinierte Ausdrücke vergleichen. Ist zum Beispiel A = 0 und B = 20, dann ergibt sich:

```
>not a and (b<50-b)
```

Eingabe eines numerischen Ausdrucks mit logischen, Vergleichs- und arithmetischen Operatoren.

```
 1
```

Der Ausdruck wird als «wahr» bewertet.

Beachten Sie, daß das Klammernpaar die Reihenfolge der Auswertung bestimmt.

Priorität von Operatoren

Bei der Auswertung eines Ausdrucks besitzen Klammerausdrücke höchste Präferenz. Die anderen Operatoren folgen entsprechend ihrer Anordnung in der folgenden Tabelle.

Rangordnung	Operation
Höchste	( ) Klammern.
↓	Funktionen.
↓	^ Exponentiation.
↓	NOT.
↓	*, /, DIV oder \.
↓	+, −, Negationsminus.
↓	=, <>, oder #, >, >=, <, <= Vergleichsoperatoren.
↓	AND.
Geringste	OR , EXOR.

Wenn ein Ausdruck zwei oder mehr Operationen der gleichen Rangordnung enthält, werden diese von links nach rechts abgearbeitet.

Behandlung mathematischer Fehler (DEFAULT ON, DEFAULT OFF)

Wenn Sie einen mathematischen Ausdruck falsch eingeben, kann ein Syntaxfehler auftreten. Wenn ein Ausdruck korrekt eingegeben wurde, kann ein mathematischer Fehler aufgrund eines nicht erlaubten Arguments oder eines nicht definierten Werts auftreten. Beim unkontrollierten Auftreten eines solchen Fehlers würde die Ausführung eines laufenden Programms angehalten werden. Der HP-75 stellt jedoch für bereichsüberschreitende Ergebnisse bei den folgenden Funktionen Ersatzwerte zur Verfügung, korrigiert somit die Fehlerbedingung und verhindert den Abbruch einer Programmausführung. Der Computer macht Sie mit einem Tonsignal und einer Warnung auf den Fehler aufmerksam. Danach stellt er einen Ersatzwert zur Verfügung und setzt die Programmausführung fort (falls nicht ON ERROR deklariert wurde, siehe Seite 258).

Bei OPTION ANGLE DEGREES gelten folgende Warnbedingungen und Ersatzwerte:

Warnungs-Nummer	Warnbedingung	Ersatzwert
1	Underflow; ein von Null verschiedenes Ergebnis zwischen −EPS und +EPS.	0
2	Overflow des Variablenbereichs: • Für INTEGER Variablen. • Für SHORT Variablen. • Für REAL Variablen.	 ±99999 ±9.9999E99 ±9.999999999E499
3	COT oder CSC gleich Unendlich; wegen eines Arguments $n \times 180°$, n ganzzahlig.	9.999999999E499
4	SEC oder TAN gleich Unendlich; wegen eines Arguments $(2n+1) \times 90°$.	9.999999999E499
5	Null hoch negativen Wert.	9.999999999E499
6	Null hoch Null.	1
7	Verwendung einer Variablen ohne zugewiesenen Wert.	0
8	Division durch Null.	9.999999999E499

Setzen Sie den Computer zum Beispiel auf OPTION ANGLE DEGREES und versuchen Sie, den Secans von 90 Grad zu berechnen:

```
>sec(90)
```

Erzeugt einen Bereichsüberlauf.

```
9.9999999999E499
```

Ein Tonsignal und eine Warnung erscheint; die Statusanzeige **Error** wird jedoch nicht gesetzt, und der HP-75 benutzt einen Ersatzwert (in diesem Fall maschinenunendlich).

Nach dem Zurücksetzen des Systems benutzt der HP-75 automatisch die obenstehenden Ersatzwerte, so daß mathematische Überlauffehler die Programmausführung nicht anhalten (auch wenn eine Warnung erscheint). Führen Sie den Befehl DEFAULT OFF aus, wenn Sie auf solche Fehler reagieren wollen.

```
DEFAULT OFF
```

Die Eingabe default off schaltet das System auf normale Fehlerverarbeitung; es werden keine Ersatzwerte für mathematische Fehler vorgegeben. Befindet sich der HP-75 im DEFAULT OFF Modus, dann erzeugt ein Berechnungsversuch von sec(90) die Fehleranzeige 4–TAN OR SEC INF, die Statusanzeige **ERROR** erscheint, und es wird kein Ersatzwert zur Verfügung gestellt. Dieser Fehler würde nun ein laufendes Programm anhalten.

Mit der folgenden Eingabe schalten Sie das System auf Ersatzwert-Vorgabe:

```
default on
```

Mit [RTN] stellen Sie den ursprünglichen Modus der Fehlerbehandlung wieder her.

Warnung 7 –no value– tritt auf, wenn Sie versuchen, eine *undefinierte* Variable (eine Variable, der Sie keinen Wert zugewiesen haben) zu benutzen.

Beispiel:

```
>b9
```

Versuch, den Wert einer nicht definierten Variablen anzuzeigen.

```
WARNING: no value
```

Anzeige dieser Meldung, danach erscheint eine 0 in der Anzeige.

Wenn der HP-75 auf DEFAULT ON geschaltet ist, benutzt er Null als Ersatzwert für undefinierte numerische Variablen, und den Nullstring (aus null Zeichen bestehend) für undefinierte Stringvariablen. Der Variablen wird jedoch *kein* Wert zugewiesen; sie bleibt undefiniert, bis sie explizit einen Wert zugewiesen bekommt.

Abschnitt 6

TIME-Modus Operationen

Inhalt

Einführung

Mit der Taste [TIME] können Sie den HP-75 in TIME-Modus schalten und den Wochentag, das Datum und die Zeit anzeigen:

```
MON 02/14/1983 01:03:15 PM
```

Typische TIME-Anzeige.

In diesem Abschnitt wird die Spezifikation dreier weiterer TIME-Anzeigeformate erläutert:

```
MON 14\02\1983 01:04:48 PM
```

TIME-Anzeige im Format Tag\Monat\Jahr.

```
MON 02/14/1983 13:04:48 **
```

Eine TIME-Anzeige im 24-Stundenformat. Die Zeit springt von 23:59:59 auf 00:00:00.

```
MON 14\02\1983 13:04:48 **
```

Ein TIME-Format im kombinierten Tag\Monat\Jahr und 24 h-Format.

Der Cursor in der TIME-Anzeige steht auf der ersten Stelle des fünfstelligen *Befehlsfeldes*:

```
MON 02/14/1983 01:04:48 PM
```

Befehlsfeld
der TIME-Anzeige.

Sie können in das Befehlsfeld fünf Befehle zur Kontrolle des Stellvorgangs, der Anzeigeformate und der Genauigkeit der Systemuhr eingeben.

SET	Stellen der Uhr.
STATS	Spezifiziert die alternativen Anzeigeformate und den Terminkalender-Modus.

ADJST	Stellt die Uhr vor bzw. nach.
EXACT	Eicht die Uhr durch Zeitmarken.
RESET	Löscht die Zeitmarken und die Anpassung der Ganggeschwindigkeit.

Diese Befehle können in Groß- oder Kleinbuchstaben in das Befehlsfeld eingegeben werden und werden beim Drücken der Taste [RTN] ausgeführt. Wenn Sie in eine beliebige Maske des TIME-Modus einen den Bereich überschreitenden Wert eingeben, erhalten Sie Fehler 89 – bad parameter. Wenn Sie ein nicht erlaubtes Zeichen in das Befehlsfeld eingeben, erhalten Sie Fehler 78 – syntax. In beiden Fällen erscheint die unrichtige Zeile noch einmal zur Korrektur.

Setzen der Uhr (SET)

Die Uhrzeit wird angezeigt bis:

- Sie den Betriebsmodus auf EDIT oder APPT ändern.
- Sie einen der Befehle SET, STATS oder ADJST im TIME- Modus ausführen (wird im folgenden besprochen).
- sich der HP-75 ausschaltet – der Computer befindet sich beim Wiedereinschalten im EDIT-Modus.

Die Uhr des HP-75 arbeitet ununterbrochen, auch bei ausgeschaltetem Gerät. Nur ein Zurücksetzen des Systems unterbricht den Gang der Uhr (siehe Abschnitt 1, «Zurücksetzen des HP-75»).

Wenn der Computer nach einem Zurücksetzen des Systems wieder eingeschaltet wird, erscheint die Zeitsetzmaske, und die Uhr beginnt bei folgenden Anfangswerten zu arbeiten:

01/01/0000	Datum: erster Januar 0000 A.D.
12:00:01 AM	Zeit: eine Sekunde nach Mitternacht.

Die TIME-Anzeige enthält diese Werte, wenn Sie direkt nach dem Zurücksetzen zweimal [ATTN] oder [ATTN] gefolgt von einem Wagenrücklauf/Zeilenvorschub drücken (Seite 43).

Wenn Sie set [RTN] in das TIME-Befehlsfeld eingeben, erhalten Sie ebenfalls die Zeitsetzmaske. Mit [ATTN] können Sie wieder auf die TIME-Anzeige zurückschalten.

Mit der Taste [TAB] können Sie Anzeigefelder überspringen, und mit der Tastenkombination [SHIFT] [TAB] können Sie zurückspringen, um die Zeitsetzmaske zu füllen. Sie können mit Ausnahme der Taste [I/R], die im TIME-Modus desaktiviert ist, jede Taste zum Ausfüllen der Zeitsetzmaske benutzen. Die Taste [CLR] zum Beispiel löscht Ihre Zeitsetzmaske. Die Felder Stunde und AM/PM sind nicht vollständig unabhängig. Wenn Sie das AM/PM-Feld ändern, *müssen* Sie in das Stundenfeld ebenfalls eine Zahl eingeben - die laufende Stunde ist mit dem gegebenen AM/PM-Zustand verknüpft. Sie können eine Zahl zwischen 13 und 23 in das Stundenfeld eingeben, wenn Sie in das AM/PM- Feld ** (zwei Sterne) anstelle von AM/PM eingeben.

Sobald Sie [RTN] drücken, wird die Uhr auf die in der Maske angezeigte Zeit eingestellt. Der Prozeß hat eine Verzögerungszeit von etwa 0.1 Sekunden. Nicht vollständig gefüllte und leere Zeitsetzfelder bleiben auf ihre derzeitigen Werte eingestellt.

Formatänderungen (STATS)

Mit dem Befehl STATS (*Status*) können Sie wählen:

- Monat/Tag/Jahr – und Tag\Monat\Jahr-Formate.
- AM/PM und **(24-Stunden) Formate.
- Normaler Jahreskalender und erweiterter Kalender zur Terminplanung.

Auf folgende Weise können Sie das Zeitformat des HP-75 in die Form Tag\Monat\Jahr und 24-Stunden-Anzeige bringen. Tasten Sie stats in das TIME-Befehlsfeld und drücken Sie [RTN].

```
MON 02/14/1983 01:10:34 PM stats
```

```
Date: Mdy, ~Time: AM, Appt: YEAR
```

Die STATS-Maske erscheint.

Die Tilde (~) vor dem Wort Time zeigt an, daß es sich um eine *ungefähre* Zeitangabe handelt, da Sie die Uhr bis jetzt noch nicht geeicht haben (Seite 95).

An dieser Stelle können Sie:

- mit dem Befehl DM (in Groß- oder Kleinbuchstaben) ein anderes Format des Datums setzen, so daß MDY zu DMY wird.
- mit der Eingabe von ** in das AM-Anzeigefeld auf 24-Stunden-Notation umschalten.
- das Wort YEAR mit EXTD (*extended search*) überschreiben. Dadurch wird die Arbeitsweise des HP-75 im APPT-Modus geändert. Das Aufsuchen von Terminen wird in Abschnitt 7, «APPT-Modus Operationen» behandelt.

Tasten Sie folgende Eingaben in die STATS-Maske:

```
Date: dmY, ~Time: **, Appt: YEAR
```

Die modifizierte STATS-Maske. Der Cursor springt jeweils in das richtige Feld.

```
MON 14\02\1983 13:15:43 **
```

D\M\Y-Format wird mit Abwärtsstrichen (\) gekennzeichnet. Der Doppelstern (**) zeigt 24-Stunden Notation an.

Wenn Sie danach den Befehl SET ausführen, erhalten Sie die folgende Anzeige:

```
MON 14\02\1983 13:16:32 ** set
```

Der Befehl SET in der TIME-Anzeige.

```
Set Dy\Mo\Year Hr:Mn:Sc **
```

Die Zeitsetzmaske enthält die soeben spezifizierten Formate.

Mit [ATTN] erhalten Sie die TIME-Anzeige zurück. Wenn Sie dann [APPT] drücken, können Sie die neue Struktur der APPT-Maske anzeigen:

```
Day Dy\Mo\Yr Hr:Mn ** #1N !Note
```

Die neuen Formate bleiben auch in den Anzeigen des APPT-Modus bestehen.

Sie können nun in diesen neuen Formaten neue Termine eingeben und alte Termine abrufen.

Schalten Sie schließlich mit [EDIT] wieder in den EDIT-Modus zurück und geben Sie catall [RTN] ein. Sehen Sie dann mit [↓] Ihre Files im Speicher durch. Sie stellen fest, daß deren Katalogeinträge im neuen Tag\Monat\Jahr-Format erscheinen. Beispiel:

```
SINGSONG B 47 09:45 05\02\83
```

Der Katalogeintrag des Files zeigt das neue D\M\Y-Format. (Kataloge benutzen immer 24-Stunden Notation.)

Neue TIME-Anzeigeformate bleiben erhalten, bis sie mit einem weiteren STATS-Befehl oder durch Zurücksetzen des Systems geändert werden. Wenn Sie wieder die ursprünglichen TIME-Anzeigeformate setzen wollen, schalten Sie mit [TIME] in den TIME-Modus und geben Sie wieder stats [RTN] ein:

```
Date: DMY, ~Time: **, Appt: YEAR
```

Die neue STATS-Maske.

Geben Sie md in das DMY-Feld ein; tasten Sie danach entweder am oder pm über den Doppelstern (**), und schließen Sie die Eingabe mit RTN ab.

```
Date: mdy, ~Time: am, Appt: YEAR
```

Ändern der STATS-Maske.

```
MON 02/14/1983 01:16:54 PM
```

Die ursprünglichen M/D/Y- und AM/PM-Formate sind wieder gesetzt.

Anpassen der Uhr (ADJST)

Mit dem Befehl ADJST können Sie *relative* Anpassungen der Uhrzeit durchführen. Sie können die Uhr um maximal 99 Stunden, 99 Minuten und 99.9 Sekunden vor- oder zurückstellen.

Beispiel: Auf dem Flug von San Francisco nach New York wollen Sie die Uhr 3 Stunden vorstellen. Geben Sie adjst in das TIME-Befehlsfeld ein und drücken Sie RTN:

```
Adjust (N) + Hr+Mn+Sc.t
```

Die ADJST-Maske erscheint.

Dieser Vorgang ist eine *Normal*anpassung, deshalb wird das Feld N nicht geändert. Um die Uhr auf eine frühere Zeit zu stellen, um also Zeit von der Anzeige zu subtrahieren, müssen Sie mit der Taste − ein Minuszeichen eingeben.

In diesem Beispiel soll die Uhr um 3 Stunden vorgestellt werden, es müssen also 3 Stunden addiert werden; deshalb muß das Pluszeichen (+) stehenbleiben – setzen Sie den Cursor mit TAB auf das Feld Hr (Stunden). Geben Sie den Änderungsbetrag ein. Jede Stelle kann eine Ziffer von 0 bis 9 aufnehmen. Nicht gefüllte Felder Hr, Mn, Sc und t (Zehntelsekunden) werden mit Null gefüllt. (Die beiden inneren Pluszeichen können nicht geändert werden.)

Drücken Sie für dieses Beispiel einmal TAB, geben Sie 03 (oder *Leerzeichen* 3 oder 3 *Leerzeichen*) ein, und schließen Sie mit RTN ab:

```
Adjust (N)  + 03+Mn+Sc.t
```

```
Mon 02/14/83 04:17:19 PM
```

Die angepaßte TIME-Anzeige.

Kalibrieren der Uhr (EXACT)

Sie können mit dem Befehl EXACT die Uhr kalibrieren. Die Geschwindigkeit der Uhr kann in einem Bereich von ±10% verändert werden.

Hinweis: Die Ganggenauigkeit der Uhr wird durch die äußere Umgebung des Computers beeinflußt. Schlagen Sie die Betriebsdaten in Anhang B nach.

Auf folgende Weise können Sie die Ganggeschwindigkeit der Uhr korrigieren, wenn sie zum Beispiel um fünf Sekunden wöchentlich vorgeht. Synchronisieren Sie zuerst die TIME-Anzeige mit Hilfe der Prozeduren SET oder ADJST mit einem genauen Zeitgeber. Geben Sie danach exact RTN ins TIME-Befehlsfeld ein:

```
MON 02/14/1983 01:22:47 exact
```

Genaue Uhrzeit.

```
MON 02/14/1983 01:22:48 PM
```

Die Einstellung bleibt unverändert, aber eine Zeitmarke wurde gesetzt.

Prüfen Sie nun die STATS-Maske:

```
MON 02/14/1983 01:22:55 PM stats
```

```
DATE: MDY, *Time: AM, Appt:YEAR
```

Der Stern vor Time bedeutet, daß zumindest einmal EXACT ausgeführt wurde.

Mehr brauchen Sie diese Woche nicht zu tun. Vergleichen Sie später die TIME-Anzeige mit der geeichten Zeitquelle. Wenn Sie feststellen, daß die Uhr um fünf Sekunden zu schnell geht, führen Sie einen weiteren ADJST-Befehl durch, geben ein Minus ein, springen auf das Sekundenfeld und geben *Leerzeichen* 5 ein:

```
Adjust (N)  - Hr+Mn+ 5.t
```

Die Maske ADJST zeigt -5 Sekunden Korrektur. Sie können auch 05 oder 5 *Leerzeichen* eingeben.

```
MON 02/21/1983 06:22:14 PM
```

Relative Anpassung um –5 Sekunden.

Sie können nun den ADJST-Prozeß wiederholen und das Zehntelsekundenfeld der TIME-Anzeige kalibrieren. Sie können die Zeitanzeige auch mit dem Befehl SET und der Zeitsetzmaske stellen.

Führen Sie nach der Synchronisation der TIME-Anzeige einen zweiten EXACT-Befehl aus:

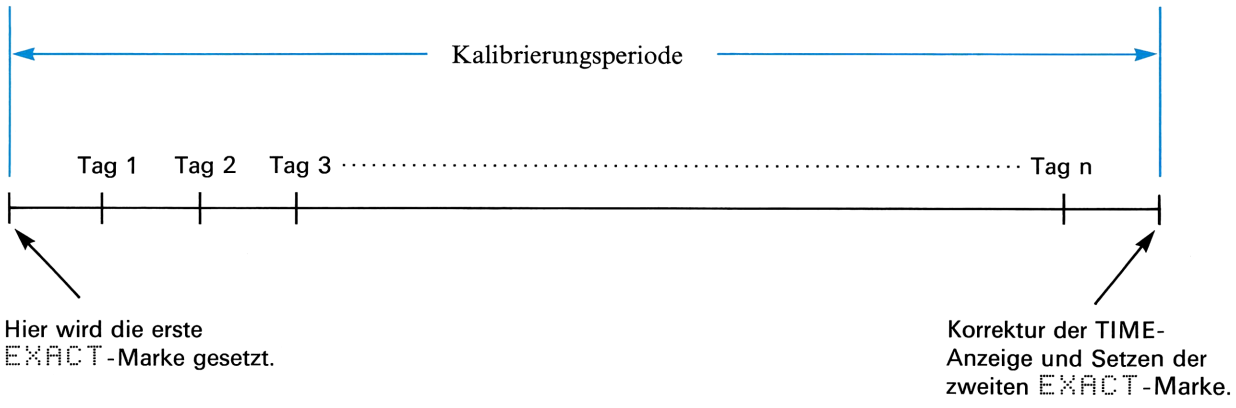
```
MON 02/21/1983 06:25:13 PM exact
```

Eingabe eines zweiten EXACT-Befehls.

```
MON 02/21/1083 06:25:14 PM
```

Markiert das Ende der Kalibrier-Periode. Die Uhr ist nun um 5 Sekunden/Woche verlangsamt.

Die Ganggenauigkeit der Uhr wird entsprechend diesem Diagramm korrigiert:



Hier eine Zusammenfassung:

1. Korrigieren Sie zu Beginn die TIME-Anzeige mit ADJST oder SET.
2. Tasten Sie exact in das TIME-Befehlsfeld und drücken Sie RTN.
3. Korrigieren Sie nach einigen Tagen, Wochen oder Monaten die TIME-Anzeige, wieder mit ADJST oder SET. Sie können die TIME-Anzeige mehrmals während der Kalibrierperiode korrigieren.
4. Führen Sie einen weiteren EXACT-Befehl aus. Damit erzeugen Sie einen Korrekturfaktor, der ab diesem Zeitpunkt die Ganggeschwindigkeit der Uhr korrigiert.

Der HP-75 berechnet den Korrekturfaktor mit Hilfe den beiden *letzten* EXACT-Befehle. Die Ausführung zweier EXACT-Befehle hintereinander ohne dazwischenliegende Korrektur mit SET oder ADJST bewirkt keine Änderung des Korrekturfaktors.

Wenn es sich bei den Korrekturen um mehr als 30 Minuten handelt, benutzt EXACT den Wert, um den die Korrektur das nächstkleinere Vielfache von 30 Minuten überschreitet – ein Zeitunterschied zwischen 0 und 30 Minuten.

Eine Korrektur nach einer kurzen Kalibrierperiode (weniger als einen Tag) erzeugt leicht eine falsche Ganggeschwindigkeit. Die Kalibrierung wird um so genauer, je länger die Kalibrierperiode ist. Bei unerlaubten Anpassungswerten erzeugt der HP-75 bei der Ausführung des zweiten EXACT-Befehls Warnung 70 – time adjust bad –, beläßt den Korrekturfaktor bei seinem ursprünglichen Wert und registriert den Befehl EXACT als Beginn einer neuen Kalibrierperiode.

Zurücksetzen der Ganggeschwindigkeit (RESET)

Der momentane Korrekturfaktor zur Ganggeschwindigkeit der Uhr bleibt erhalten bis:

- Sie einen weiteren EXACT-Befehl eingeben.
- Sie den Befehl RESET eingeben. Tasten Sie reset ein, und drücken Sie RTN.

Nach Ausführung eines RESET-Befehls sind die EXACT-Marken und der Korrekturfaktor gelöscht. Die STATS-Maske zeigt wieder ~TIME an Stelle von *TIME.

Normal- und Absolutanpassung (N, A)

Bei der Einstellung N (*normal*) in der ADJST-Maske wird ein bestimmter Anteil der Geschwindigkeitsanpassung zur Fehlerkorrektur verwendet. Dieser Anteil wird so berechnet, daß Zeitzonen, Sommerzeit und andere halb- oder ganzstündige Korrekturen die Ganggeschwindigkeit der Uhr nicht beeinflussen. «Übriggebliebene» Zeitbeträge – bis zu 15 Minuten bei jeder Anpassung – werden jedoch als Korrekturwerte betrachtet, wenn die ADJST-Maske als Teil des Kalibriervorgangs benutzt wird.

Die Einstellung A (*absolut*) in der ADJST-Maske bedeutet, daß die Anpassung vollständig auf einem Wechsel der Zeitzone beruht und nicht zur Fehlerkorrektur benutzt wird.
Geben Sie adjst RTN in das TIME-Befehlsfeld ein, wenn Sie eine absolute Anpassung durchführen wollen. Drücken Sie dann ← oder BACK und geben Sie ein a oder A in die ADJST-Maske ein:

```
Adjust (a)  + Hr+Mn+Sc.t
```

Umstellen der ADJST-Maske auf Absolut-Anpassung.

Absolute Anpassungen sind selten erforderlich; Sie können mit der Maske A kleine Korrekturen anbringen, ohne die Ganggeschwindigkeit der Uhr zu beeinflussen.

Zeit- und Datumsfunktionen (DATE, TIME, DATE$, TIME$)

Sie können im EDIT-Modus mit vier programmierbaren Funktionen auf die Uhr des HP-75 zugreifen: DATE, TIME, DATE$ und TIME$.

Die Funktion DATE gibt einen ganzzahligen Wert der Form *yyddd* zurück, der das Datum repräsentiert.

Beispiel:

```
>date
```

```
83038
```

Numerische Darstellung des 7. Februar 1983

Die linken beiden Ziffern enthalten die Jahreszahl im laufenden Jahrhundert. Die rechtsstehenden drei Ziffern bedeuten die Nummer des Tages im laufenden Jahr; der erste Januar hat zum Beispiel die Nummer 001.

Die Funktion TIME gibt die Anzahl Sekunden zurück, die seit der letzten Mitternacht verstrichen sind, d.h. einen Wert zwischen 0 (um Mitternacht) und 86399.999 (um 23:59 und 59.999 Sekunden). Die Zeitwerte sind auf Millisekunden gerundet.

Beispiel:

```
>time
```

```
49272.555
```

Numerische Darstellung von 13:41:12 und .555 Sekunden

Mit der Funktion TIME können Sie Rechenzeiten bestimmen. Die Differenz der Zeitwerte vor und nach einem Programmlauf entspricht der für die Programmausführung benötigten Zeit. Ein Beispiel dazu finden Sie auf Seite 220 (Stoppuhrprogramm).

Die Funktion DATE$ gibt einen Achtzeichenstring im Format *yy/mm/dd* (Jahr/Monat/Tag) zurück.

Beispiel:

```
>date$
```

```
83/02/14
```

Stringdarstellung des 14.Februar 1983.

Die Funktion TIME$ gibt einen Achtzeichenstring im Format *hh:mm:ss* (Stunde:Minuten:Sekunden) mit Werten zwischen 00:00:00 und 23:59:59 zurück.

Beispiel:

```
>time$
```

```
13:41:52
```

Stringdarstellung von 13:41:52 Uhr.

Abschnitt 7

APPT-Modus Operationen

Inhalt

Einführung

In diesem Abschnitt werden folgende Operationen behandelt:

- Einrichten von Terminen, um Meldungen anzuzeigen und Befehle auszuführen.
- Überprüfen und Editieren der im appt-File gespeicherten Termine.
- Bestätigen und Bearbeiten fällig gewordener Termine.
- Ausdehnen der Terminspeicherkapazität Ihres Computers durch Verwendung von Massenspeichern.
- Einrichtung einer Vielzahl von einmaligen und wiederholenden Terminen mit zwei Kalendern – dem Jahreskalender und dem erweiterten Kalender.
- Ein- und Ausschalten des APPT-Modus.
- Suchen von Tag/Datumspaaren im erweiterten Kalender.

Einrichten von Terminen ([APPT])

Termine werden auf dem HP-75 im APPT-Modus eingegeben: Drücken Sie [APPT]. Falls kein Termin fällig ist, d.h. falls die Statusanzeige **APPT** nicht gesetzt ist, erscheint die APPT-Maske:

```
Day Mo/Dy/Yr Hr:Mn AM #1N !Note
```

Mit der APPT-Maske können Sie Termine einplanen.

Füllen Sie soviel in der APPT-Maske aus wie erforderlich. Der HP-75 liefert Ersatzwerte für nicht benutzte Wochentag-, Datums- und Zeitfelder:

APPT-Maske	Felder	Ersatzwerte
Wochentag	Day	Der Tag (SUN–SAT), aus dem Datum berechnet.
Datum	Mo	Der laufende Monat oder der nächstmögliche Monat vor Ablauf des nächsten Jahres.
	Dy	Der heutige Tag oder der nächste mögliche Tag.
	Yr	Wenn möglich das laufende Jahr; andernfalls das folgende Jahr. Termine können beliebig innerhalb der nächsten 365 (366) Tage eingerichtet werden. Das Feld Yr wird nur benutzt, wenn geplante Termine außerhalb dieses Bereichs liegen. (Siehe erweiterter Kalender, Seite 110.)
Zeit	Hr	Die laufende oder die folgende Stunde, falls der Termin am gleichen Tag fällig wird, andernfalls 00:00 Uhr.
	Mn	Die folgende Minute, falls der Termin zur gleichen Stunde fällig wird; andernfalls 00.
	AM	AM, falls der Termin nicht am laufenden Tag fällig wird.
Alarmart	#1	Ein Alarm des Typs 1, ein kurzer Piepton.
Terminart	N	Ein *normaler* oder Typ N Termin, der nur einmal fällig wird.
Notiz- oder Befehlsymbol	!	Beginn des Notizfeldes. (Kann in > geändert werden, um einen BASIC-Befehl aufzunehmen.)
Notiz- oder Befehlsfeld	Note	Ein nicht ausgefülltes Feld Note verschwindet im vollständigen Termin.

Allgemein werden nicht ausgefüllte Datums- und Zeitfelder auf die nächstmöglichen *zukünftigen* Werte eingestellt.

Die Felder Hr und AM/PM sind nicht vollständig voneinander unabhängig. Wenn Sie einen Eintrag ins AM/PM-Feld machen, müssen Sie ebenfalls eine Zahl ins Stundenfeld eintragen – die laufende Stunde ist mit dem gegebenen AM/PM-Wert verknüpft.

Benutzen Sie das Alarm-Feld (#1), um die gewünschte Alarmart einzustellen, die den Termin ankünden soll. Sie haben 10 Arten zur Auswahl, 0 bis 9.

Ziffer	Alarmart
0	Kein Tonsignal, der Piepton wird unterdrückt.
1	Ein kurzes Zirpen.
2	Ein langer, tiefer Ton
3	Ein dreimal wiederholtes Zweitonsignal.
4	Eine Folge von hohen, eindringlichen Tönen.
5	Ein langer tiefer Ton, gefolgt von einem langen hohen Ton.
6	Eine Folge von acht Sirenenklängen.
7	Ein alle 15 Sekunden wiederholter Alarm des Typs 2.
8	Ein alle 15 Sekunden wiederholter Alarm des Typs 4.
9	Ein alle 15 Sekunden wiederholter Alarm des Typs 6.

Die Alarmarten 7, 8 und 9 wiederholen ihr Signal alle 15 Sekunden, bis Sie eine Taste drücken. Sie sollten diese Daueralarmtypen nur setzen, wenn der Computer vom Netz betrieben wird, weil sie, über längere Zeit aktiv, den Batteriesatz entladen könnten. Abfallende Betriebsspannung bewirkt nach einiger Zeit ein Einstellen aller Operationen, einschließlich der APPT-Funktionen, und ein Ausschalten des Geräts. (Siehe Anhang B.)

Mit dem Befehl BEEP OFF werden alle Alarmsignale *außer* den Typen 6 und 9 desaktiviert.

Im Feld N können Sie drei Terminarten spezifizieren:

Symbol	Art des Termins
N	Ein *normaler* oder einmaliger Termin, dessen APPT-Statusanzeige gesetzt bleibt, bis Sie im APPT-Modus die Taste [ATTN] drücken.
R	Ein selbstplanender, sich *wiederholender* Termin, der sich beim Ertönen des Alarms selbst aktualisiert.
A	Ein selbstplanender Termin, den Sie vor seiner Erneuerung mit [ATTN] *bestätigen* müssen.

Weitere Information finden Sie auf Seite 106, «Terminarten».

Das Notiz/Befehlssymbol (!) kennzeichnet den Beginn des Note- Feldes, das bis zu 68 Zeichen Text oder beliebige Kombinationen von Befehlen des EDIT-Modus und BASIC-Anweisungen enthalten kann. Siehe auch Seite 108.

Die Tasten [TAB] und [TAB] [SHIFT] lassen den Cursor vorwärts und rückwärts über die Anzeigefelder springen, zusätzlich können auch die Tasten [←], [SHIFT] [→] usw. die Eingabe beschleunigen.

Drücken Sie nach der Eingabe und Prüfung Ihres Termins [RTN]. Der vollständige Termin wird mit den eingesetzten fehlenden Werten angezeigt – so wie er im Speicher abgelegt wird.

Mit [CLR] erhalten Sie wieder die nicht ausgefüllte APPT-Maske:

```
Day Mo/Dy/Yr Hr:Mn AM #1 !Note
```

Eingabebereit für einen neuen Termin.

Die Taste [↓] bringt den zuletzt eingegebenen Termin in die Anzeige zurück.

Das Feld DAY ermöglicht Ihnen, wöchentliche und monatliche Information zu spezifizieren:

Spezifizierter Wochentag:	Termin wird fällig:
SUN, MON,...SAT	Wenn möglich, an einem Tag der nächsten Woche; andernfalls am nächsten passenden Zeitpunkt.
SU+, MO+, ...SA+	Am nächsten Tag nach dem derzeitigen oder angegebenen Datum.
SU1, SU2, SU3... ...SA3, SA4, SA5.	Am Tag, der in die erste bis 5. Woche des laufenden oder folgenden Monats fällt.
SU–, MO–, ...SA–	Am ersten möglichen Tag vor dem angegebenen Datum. Falls vor dem heutigen Datum geplant, wird der Termin sofort beim Drücken von [RTN] angekündigt.

Wenn Sie unpassende Information in ein Feld des APPT-Modus eingeben, können Sie eine Fehlerbedingung erhalten. Wenn Sie zum Beispiel einem gegebenen Datum den falschen Wochentag zuordnen, erhalten Sie beim Drücken von [RTN] den Fehler 72 –day/date mismatch–. Die fehlerhafte Maske erscheint korrigierbereit in der Anzeige. Mit [SHIFT] [FET] können Sie die Fehlermeldung nochmals anzeigen. Löschen Sie den Fehler mit [ATTN], [APPT] oder [CLR] (oder [EDIT] oder [TIME]), oder korrigieren Sie die Anzeige mit den Editiertasten. Mit errn [RTN] erhalten Sie die Fehlernummer des zuletzt aufgetretenen Fehlers.

Der HP-75 nimmt keine zwei identischen Termine an. Zumindest ein Zeichen in einem ihrer Felder muß verschieden sein. Wenn zwei verschiedene Termine zur gleichen Zeit fällig werden, wird der zuerst eingegebene Termin zuerst angekündigt; der Alarm des zweiten Termins folgt unmittelbar darauf.

Der appt-File ([↑], [↓], CAT APPT, PURGE APPT)

Ihre Termine werden im appt-File des HP-75 abgelegt. Durch Drücken von [APPT] stellen Sie den Filepointer auf den appt-File und positionieren ihn auf den fälligen Termin. Wenn zumindest ein Termin fällig ist, wird der zuerst fällig gewordene Termin angezeigt (fällige und überfällige Termine erscheinen mit unterstrichenem Wochentag, Datum und Zeitfeld).

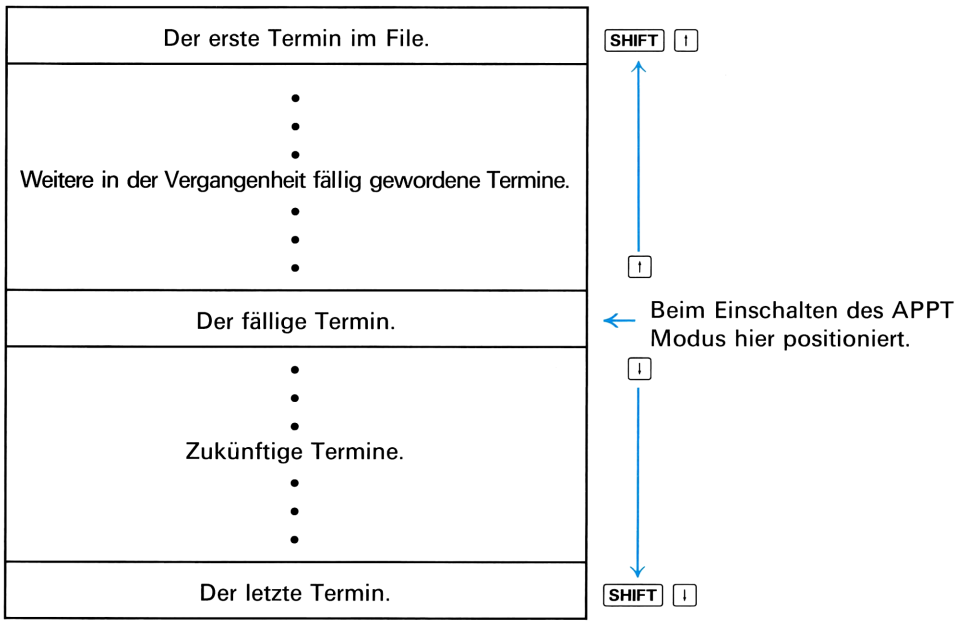
Wenn derzeit keine Termine fällig sind, wird die APPT-Maske angezeigt.

Mit den Tasten [↑] und [↓] können Sie einzelne Einträge im appt-File aufsuchen. [↑] zeigt die früheren und [↓] zeigt die späteren Termine. Bei angezeigter APPT-Maske und keinen fälligen Terminen (die Statusanzeige **APPT** erscheint nicht) gilt:

- Mit [↑] werden vergangene Termine angezeigt, die jüngsten zuerst.
- Mit [↓] werden zukünftige Termine angezeigt, die nächstfälligen zuerst. Wenn keine zukünftigen Termine anstehen, wird der letzte vergangene Termin angezeigt.

Mit [CLR] erhalten Sie wieder die APPT-Maske.

[SHIFT] [↑] zeigt den ersten, [SHIFT] [↓] den letzten Termin im File an.



Drücken Sie [EDIT] und führen Sie den Befehl CAT APPT aus, um den Katalogeintrag des appt-File zu erhalten:

```
>catappt
```

Ein Befehl im EDIT-Modus. Stattdessen können Sie auch CAT ALL verwenden. Groß/Kleinschrift und Zwischenräume spielen keine Rolle.

Drücken Sie [RTN]:

```
appt          A   21 09:40 02/05/83
```

Für einen appt-File typischer Katalogeintrag.

Filename kleingeschrieben: deutet an, daß der appt-File vom HP-75 erzeugt und bearbeitet wird.

Uhrzeit und Datum des Files – der ersten Eingabe eines Termins nach einem Löschen des vorhergehenden appt-Files.

Länge in Bytes.

Filetyp. Es können sich mehrere Terminfiles (Typ A) im Speicher befinden, aber nur einer kann den Namen appt tragen.

Sie können einen Terminfile zwar nicht listen (LIST), editieren (EDIT) oder abarbeiten (RUN), aber Sie können ihn kopieren (COPY) und löschen (PURGE).

Beispiel:

```
>purge appt
```

Mit [RTN] wird der appt-File gelöscht, sein Name aus dem Systemkatalog entfernt.

Der appt-File kann auch mit dem Befehl RENAME umbenannt werden. Die Termine in einem umbenannten Terminfile werden vom HP-75 nicht überwacht, solange der File nicht wieder in appt zurück umbenannt wird. Unterdessen erzeugt der HP-75 bei der nächsten Eingabe eines Termins einen neuen appt-File.

Wenn Sie einen Terminfile auf den appt-File kopieren, werden alle Termine beider Files in den appt-File eingeordnet.

Der Name des aktiven Terminfiles, appt, erscheint in allen Filebefehlen des EDIT-Modus ohne Anführungszeichen.

Löschen einzelner Termine ([SHIFT] [DEL])

Alle Termine bleiben im appt-File, bis Sie sie einzeln oder den ganzen File löschen. Ein einzelner Termin wird wie folgt gelöscht:

1. Schalten Sie mit [APPT] in den APPT-Modus und auf den appt- File.
2. Suchen Sie den unerwünschten Termin mit [↑] und [↓] auf.
3. Löschen Sie den Termin, sobald er in der Anzeige steht, mit [SHIFT] [DEL].

Danach wird der dem gelöschten Termin im appt-File folgende Termin angezeigt. Wenn Sie wieder [SHIFT] [DEL] drücken, werden der angezeigte Termin und nachfolgende Termine einer nach dem anderen gelöscht, sowie sie in der Anzeige erscheinen.

Sobald auf diese Weise alle Termine gelöscht sind, verschwindet der File appt aus dem Systemkatalog.

Korrektur schon geplanter Termine ([RTN], [SHIFT] [DEL])

Sie können einen schon eingegebenen Termin korrigieren oder ändern, indem Sie ihn mit [↑] oder [↓] zur Anzeige bringen und dann mit der neuen Information überschreiben. Drücken Sie danach [RTN]; der neue Termin wird im appt-File gespeichert, *ohne* daß das Original gelöscht wird.

Die Tastenkombination [SHIFT] [DEL] selbst löscht den angezeigten Termin. Wenn Sie jedoch einen Termin in der Anzeige überschreiben und *danach* [SHIFT] [DEL] drücken, *ersetzen* Sie den vorhergehenden Termin durch den neuen Termin.

Beispiel:

```
SAT 02/05/83 09:25 AM #3N !Anruf
```

Ein alter Termin soll durch einen neuen Termin ersetzt werden, z.B. am Donnerstag zur gleichen Zeit (angenommen, heute ist Dienstag, der 8. Februar).

```
thu 02/  /83 09:25 AM #3N !Anruf
```

Geben Sie thu ins Feld des Wochentags ein, überspringen Sie das Monatsfeld und schreiben Sie zwei Leerzeichen ins Tagesfeld. Die Unterstreichung wird bei der Eingabe entfernt.

Drücken Sie dann [SHIFT] [DEL]:

```
THU 02/10/83 09:25 AM #3N !Anruf
```

Der neue Termin ersetzt den ursprünglichen.

Wenn beim Drücken von [RTN] oder [SHIFT] [DEL] ein Fehler auftritt, bedeutet dies, daß die Information in der Maske einen Wochentag, ein Datum oder eine Zeit enthält, die der HP-75 nicht verarbeiten kann. Sowohl vergangene wie auch zukünftige Termine können mit [RTN] und [SHIFT] [DEL] korrigiert werden.

Sie können mit [SHIFT] [DEL] auch einen fälligen Termin, der in der Anzeige steht, *bestätigen*. Damit wird der Termin gleichzeitig bestätigt und gelöscht.

Bearbeitung fälliger Termine ([RUN])

Termine können bei eingeschaltetem und ausgeschaltetem HP-75 eintreffen. In Abschnitt 1 haben Sie gesehen, daß die Statusanzeige APPT und der eingestellte Alarmton in beiden Fällen eingeschaltet werden.

Bei eingeschaltetem HP-75 unterbricht der fällige Termin die derzeitige Operation des Computers, sei es die Abarbeitung eines Programmes oder die Anzeige der Zeit, nicht. Die Statusanzeige APPT bleibt solange gesetzt, wie die Anzeige eingeschaltet ist, um den fälligen Termin anzudeuten.

Unabhängig davon, ob der HP-75 ein- oder ausgeschaltet ist, müssen Sie, sobald vor der Bestätigung fälliger Termine mehr als ein weiterer Termin fällig wird, die Termine mit [ATTN] oder [SHIFT] [DEL] einzeln bestätigen.

Wenn ein fälliger Termin eine Meldung oder einen Befehl enthält, können Sie mit [RUN] im APPT-Modus diese Meldung anzeigen oder diesen Befehl ausführen. Wenn Sie zu *beliebiger* Zeit im APPT-Modus [RUN] drücken, wird jeder fällige Termin bearbeitet.

Beispiel:

```
TUE 02/08/83 3:00 PM #5N !Ruf
                              APPT
```

Ein fälliger Termin mit einer Meldung. Bearbeiten Sie ihn mit [RUN].

```
!Ruf zuhause an.
                              APPT
```

Die Meldung wird im EDIT-Modus für etwa 5 Sekunden, oder bis Sie eine Taste drücken, angezeigt.

```
TUE 02/08/83 3:00 PM #5N !Ruf
                              APPT
```

Wenn Sie nach 5 Sekunden noch keine Taste gedrückt haben, erscheint der fällige Termin wieder in der Anzeige und wartet auf Bestätigung.

Fällige Termine können nur *einmal* bearbeitet werden (zu jedem anderen Zeitpunkt wirkt die Taste [RUN] im APPT-Modus immer wie [APPT]).

Wenn ein Termin bei ausgeschaltetem HP-75 eintrifft, wird der Termin sofort bearbeitet. Die Statusanzeige APPT wird gesetzt, der Alarm erklingt, der HP-75 schaltet in den EDIT-Modus und die Meldung des Termins wird angezeigt, oder dessen Befehl ausgeführt. Nach ungefähr 5 Sekunden schaltet sich der HP-75 wieder aus.

Fällige Termine können Tastenfeldoperationen nicht unterbrechen. Wenn ein fälliger Termin, der eine Meldung oder einen Befehl enthält, beim Ausschalten des HP-75 noch immer nicht bearbeitet wurde, behandelt der HP-75 den Termin so, wie er ihn im Ausschaltzustand behandeln würde. Wenn Sie zum Beispiel [SHIFT] [ATTN] drücken oder den Befehl BYE ausführen, um den HP-75 auszuschalten, werden fällige Termine (bestätigte und unbestätigte) sofort bearbeitet.

Entsprechend unterbrechen fällige Termine auch nicht die Ausführung eines Programms, selbst dann nicht, wenn die Programmausführung angehalten ist – der Alarm wird jedoch eingeschaltet. Bei Ausführung des Befehls BYE innerhalb eines Programms wird der HP-75 ausgeschaltet, ohne vergangene fällige Termine verarbeitet zu haben. Wird er auf diese Weise ausgeschaltet, reagiert er nicht auf weitere fällig werdende Termine, da das Programm noch nicht abgeschlossen wurde. Beim Wiedereinschalten des Computers mit [ATTN] wird das Programm beendet. Überfällige Termine werden beim nächsten Ausschalten des HP-75 bearbeitet.

Ausschalten des APPT-Modus (ALARM OFF, ALARM ON)

Der Befehl ALARM OFF verhindert, daß zukünftige Termine fällig werden. Drücken Sie [EDIT] und geben Sie folgende Sequenz ein, um den APPT-Modus auszuschalten:

```
>alarm off
```

Drücken Sie [RTN], und zukünftige Termine werden ignoriert.

Solange ALARM OFF aktiv ist, werden bei neu eintreffenden Terminen weder die Statusanzeige APPT gesetzt noch der Alarm eingeschaltet. Der HP-75 zeigt auch nicht die entsprechenden Meldungen an oder führt deren Befehle durch. Der HP-75 *bearbeitet* jedoch überfällige Termine und läßt Sie neue Termine einrichten.

> **Wichtig:** Schalten Sie den HP-75 auf ALARM OFF, wenn Sie den Batteriesatz entfernen wollen, während der Computer nicht ans Netz angeschlossen ist. Andernfalls bedingt ein eintreffender Termin ein Zurücksetzen des Systems, wenn sich der HP-75 ohne Stromquelle einzuschalten versucht.

Mit ALARM ON im EDIT-Modus können Sie den HP-75 wieder auf normale Terminverarbeitung schalten. Danach setzen alle seit dem letzten ALARM OFF eingetroffenen Termine die Statusanzeige **APPT** und erzeugen die entsprechenden Alarmsignale. Sie können diese bestätigen und bearbeiten.

Terminarten (N, R, A, [SHIFT] [APPT])

Sie können im Feld N der APPT-Maske drei verschiedene Terminarten setzen, N, R und A.

```
Day Mo/Dy/Yr Hr:Mn AM #1N !Note
```

Das die Terminart spezifizierende Feld.

Hier ist ein Vergleich ihrer Funktion:

Art des Termins	N	R	A
Abkürzung für	normal	reschedule (sofortige Neueinplanung)	acknowledgement (Neueinplanung nach Bestätigung)
Setzen der Statusanzeige APPT	ja	kurz	ja
Tonsignal	ja	ja	ja
Bestätigung erforderlich	ja	nein	ja
Zukünftige Wiederholung	nein	ja	ja, nach Bestätigung

Eine 0 im Alarmfeld unterdrückt den Alarm. Bei einer 7, 8 oder 9 wird das Alarmsignal solange wiederholt, bis Sie eine Taste drücken.

Wenn Sie wollen, daß ein Termin sich selbst wiederholt, tasten Sie ein r oder ein a in das Feld N (Art des Termins).

Beispiel:

```
Sun Mo/Dy/Yr 7:55 PM #4r !Nova
```

Eingabe von r setzt eine Selbstwiederholung des Termins.

```
Rept=Mo+Dy+Hr+Mn | DOW
```

Die Rept-Maske erscheint.

Mit [ATTN] oder [APPT] können Sie den Vorgang abbrechen. (Sie erhalten die Fehlermeldung 76 – bad rep field – wenn Sie die Einrichtung einer Terminwiederholung abbrechen.) [CLR] löscht die Rept-Maske.

Die Eingabe in die Rept-Maske bestimmt das Datum und die Uhrzeit der Wiederholung des Termins.

Beispiele:

```
Rept=Mo+07+Hr+Mn | DOW
```

Der Termin wird jede Woche wiederholt. Nicht ausgefüllte Rept-Felder werden auf Null gesetzt.

```
Rept=Mo+Dy+02+Mn | DOW
```

Der Termin wird alle zwei Stunden wiederholt.

```
Rept=Mo+Dy+Hr+15 | DOW
```

Der Termin wird alle 15 Minuten wiederholt.

Die Felder können Werte zwischen 00 und 99, jedoch keine negativen Werte, annehmen. Beenden Sie die Eingabe der Terminwiederholungen mit [RTN].

Beispiel:

```
Rept=Mo+ 7+Hr+Mn | DOW
```

Eingabe eines wöchentlichen Alarms.

```
SUN 02/13/83 07:55 PM #4R !Nova
```

Der vollständige Wiederholungstermin.

Drücken Sie [SHIFT] [APPT] und halten Sie die Taste [APPT] gedrückt, wenn Sie das Wiederholungsintervall eines R- oder A-Termins überprüfen wollen.

Beispiel:

```
Yr=1983 | Rept=Mn+07+Hr+Mn | DOW
```

Mit [SHIFT] [APPT] können Sie kurzzeitig das Jahr und Wiederholungsintervall des angezeigten R- oder A-Termins ablesen.

Wenn ein Termin des Typs R eintrifft, erklingt der Alarm, und der Termin wird im appt-File entsprechend seinem Wiederholungsintervall neu eingeplant. Um einen A-Termin zu erneuern, müssen Sie ihn bestätigen, d.h. die Taste [ATTN] drücken, wenn er in der Anzeige steht. Ein Termin des Typs A wird aufgrund der Zeit und des Datums des Termins, nicht seiner Bestätigung, erneuert. Sowohl R- als auch A-Termine werden beliebig oft erneuert – löschen Sie sie mit [SHIFT] [DEL].

Wenn Sie den appt-File untersuchen, werden Sie feststellen, daß der HP-75 nur die derzeitige oder erneuerte Form eines zu wiederholenden Termins speichert.

Das Feld DOW in der rept-Maske ermöglicht Ihnen, Wiederholungsintervalle für Wochentagsvorgaben zu setzen.

Wenn Sie SUN, MON,..., SAT eingeben, wird der Termin auf den spezifizierten Wochentag erneuert.

Beispiel:

```
Rept=Mo+Dy+Hr+Mn | mon
```

Der zu wiederholende Termin wird auf jeden Montag geplant.

Wenn Sie SU+, MO+... in das Feld des Wochentags eingeben, erhalten Sie die gleichen Wiederholungsintervalle wie bei SUN, MON... .

Wenn Sie SU–, MO–... eingeben, wird der neue Termin auf den ersten Wochentag *vor* dem in den Monats- und Tages-Wiederholungsfeldern spezifizierten Zeitpunkt eingerichtet.

Beispiel:

```
Rept= 1+Dy+Hr+Mn | we-
```

Legt die Wiederholung des Termins auf den letzten Mittwoch vor dem Ende des einmonatigen Wiederholungsintervalls.

Wenn Sie SU1, SU2 ... SA4, SA5 eingeben, wird der Termin am spezifizierten Wochentag in der gegebenen Anzahl Wochen (erste bis fünfte) des laufenden Monats (wenn möglich) und der folgenden Monate wiederholt.

Beispiel:

```
Rept=Mo+Dy+Hr+Mn | th1
```

Der Termin wird jeden ersten Donnerstag der folgenden Monate wiederholt.

Mit einer Kombination von Feldern der Rept-Maske können Sie eine Vielzahl von Wiederholungsintervallen bestimmen.

Beispiel:

```
Rept= 1+ 7+Hr+Mn | fr2
```

Der Termin soll am zweiten Freitag jedes zweiten Monats wiederholt werden. (Die 7 zusätzlichen Tage bewirken das Überspringen des ersten Monats.

In jedem Fall wird der vollständige Termin mit RTN im appt-File abgespeichert. Mit SHIFT APPT können Sie kurzzeitig das Jahr und Wiederholungsintervall des momentanen Termins abrufen.

Befehlstermine (←, >)

Sie können Termine einrichten, die Befehle des EDIT-Modus und BASIC-Anweisungen ausführen und Rechnerausdrücke berechnen.* Ändern Sie dazu das Notizfeld in ein Befehlsfeld, indem Sie das Zeichen ! durch eine BASIC-Eingabeaufforderung (>) ersetzen.

```
Day Mo/Dy/Yr Hr:Mn AM #1N >Note
```

Springen Sie mit ← über das !. Drücken Sie dann SHIFT ·, um die BASIC-Eingabeaufforderung zu erzeugen.

Tasten Sie in das neu erzeugte Befehlsfeld einen Befehl, eine Anweisung oder einen Rechnerausdruck ein, und drücken Sie RTN, um den Termin abzuspeichern.

Beispiel:

```
ay Mo/Dy/Yr Hr:Mn AM #1N >plist
```

Eingabe des Befehls PLIST ins Befehlsfeld.

```
TUE 02/08/83 03:15 #1N >plist
```

Ersatzwerte werden eingefügt; der Befehlstermin ist eingerichtet.

Jeder Termin kann mehrere Befehle und Anweisungen enthalten. Verbinden Sie die Befehle und Anweisungen mit dem Symbol @.

Beispiel:

```
Hr:Mn AM #1N >pwidth 24 @ plist
```

Dieser Termin enthält sowohl den Befehl PWIDTH als auch den Befehl PLIST.

* Eine BASIC-Anweisung muß über das Tastenfeld ausgeführt werden können, um in einem Terminbefehl sinnvoll zu sein.

Das Befehlsfeld wird im appt-File genauso gespeichert, wie Sie es eingeben; der HP-75 ändert Ihre Zwischenräume, Kleinbuchstaben und Abkürzungen nicht.

Der HP-75 verarbeitet fällige Termine, die Befehlsfelder enthalten, nicht anders als Termine mit Notizfeldern. Ein fälliger Termin unterbricht den Betrieb des HP-75 bei eingeschaltetem Rechner nicht. Stattdessen wird das Befehlsfeld eines fälligen Termins unter einer dieser Bedingungen abgearbeitet:

- Wenn Sie im APPT-Modus [RUN] drücken.
- Wenn Sie in einem beliebigen Modus [SHIFT] [ATTN] drücken.
- Wenn Sie den Befehl BYE über das Tastenfeld ausführen.
- Wenn sich der HP-75 bei STANDBY OFF nach 5 Minuten ohne erfolgte Eingabe ausschaltet.
- Wenn ein Termin bei ausgeschaltetem Rechner fällig wird.

In allen genannten Fällen – mit Ausnahme des ersten – schaltet sich der HP-75 nach Abarbeitung eines Termins selbst wieder aus. Folglich kann ein ins Kommandofeld eingegebener Befehl STANDBY ON den Rechner nicht eingeschaltet erhalten, sondern nur STANDBY ON setzen.

Alle drei Terminarten – N, A und R – können Befehlsfelder enthalten. Nehmen wir an, es sei ein kurzes Programm mit Namen WECKER im Speicher, das eine Weckmelodie spielt. So können Sie dieses Programm jeden Morgen um 7:00 ablaufen lassen:

```
Day Mo/Dy/Yr 07:Mn AM #Or >Note
```

Setzen eines sich wiederholenden Termins (Typ R). Die Null unterdrückt den Tonalarm.

Geben Sie dann run, gefolgt vom entsprechenden Filenamen, ein:

```
y/Yr 07:Mn AM #Or >run 'wecker'
```

Setzt die Ausführung des Programms WECKER auf 7:00 morgens.

Beachten Sie, daß der Filename in Anführungszeichen eingeschlossen sein muß. Mit [RTN] erscheint die Rept-Maske. Füllen Sie sie aus und drücken Sie [RTN] ein zweites Mal:

```
Rept=Mo+01+Hr+Mn | DOW
```

Setzen des täglichen Programmstarts.

```
WED 02/09/83 07:00 AM #OR >run
```

Der vollständige Termin, auf 7:00 des nächsten Morgens eingerichtet, wird angezeigt.

Neue APPT-Masken (STATS)

Der Befehl STATS – ein TIME-Modus Befehl – steuert das Format der TIME- und der APPT-Anzeige. Sie erhalten die STATS-Maske, wenn Sie stats [RTN] in das TIME-Befehlsfeld eingeben:

```
TUE 02/08/1983 04:11:23 PM stats
```

Die TIME-Maske mit dem Befehl stats.

```
Date: MDY, ~Time: AM, Appt: YEAR
```

Die STATS-Maske erscheint.

Tasten Sie dm, um ein Tag\Monat\Jahr Format zu spezifizieren. Die Eingabe ** in das AM/PM Feld bewirkt das Setzen der 24-Stunden-Notation.

```
Date: dmY, ~Time: **, Appt: YEAR
```

Spezifizieren eines neuen Formats für Datum und Zeit. Der Cursor springt auf die entsprechenden Felder.

```
TUE 08\01\1983 16:12:52 **
```

Die modifizierte TIME-Anzeige.

Die STATS-Maske bestimmt das Eingabeformat Ihrer Termine im APPT-Modus. Drücken Sie [APPT]:

```
Day Dy\Mo\Yr Hr:Mn ** #1N !Note
```

Geben Sie das Datum und die Zeit neuer Termine dem neuen Format entsprechend ein.

Wenn Sie 24-Stunden-Notation verwenden, können Sie die Zeit eines neuen Termins auf zwei Weisen eingeben:

- Eingabe der Zeit als Zahl zwischen 00:00 und 23:59. (Im AM/PM-Format kann 24-Stunden-Notation verwendet werden, wenn Sie das AM-Feld mit ** überschreiben.)
- Eingabe der Zeit in der AM/PM-Form. 17:30 wird zum Beispiel als 05, 30 und pm eingegeben. Der HP-75 wandelt diese Zeit in das 24-Stunden-Format um.

Bestehende Termine im appt-File werden entsprechend dem gängigen STATS-Format angezeigt.

Hinweis: Wenn Sie die Uhr in TIME-Modus stellen, bleiben Ihre Termine mit den ursprünglichen Zeit- und Datumsinformationen erhalten. Aufgrund der neuen Uhrzeit können dann Termine fällig werden.

Der erweiterte Kalender (EXTD)

Nach einem Zurücksetzen des Systems verwendet der HP-75 einen Jahreskalender; Termine können für jeden Zeitpunkt innerhalb der nächsten 365 (366) Tage eingerichtet werden. Wenn Sie in das Feld YEAR der STATS-Maske EXTD eingeben, wird der Kalender des APPT-Modus in einen *erweiterten* Kalender geändert; Sie können dann Termine vom Jahr 0000 bis 9999 planen. Drücken Sie [TIME], tasten Sie stats [RTN] ein, geben Sie extd in das Jahresfeld ein und drücken Sie [RTN]:

```
Date: mdy, ~Time: am, Appt: extd
```

Die STATS-Maske, die Datums- und Zeitformate sowie den erweiterten Kalender spezifiziert.

```
TUE 02/08/1983 04:20:56 PM
```

Die geänderte TIME-Anzeige.

Drücken Sie dann [APPT]: Die APPT-Maske sieht gleich aus, aber das Feld Yr besitzt nun eine besondere Bedeutung.

Benutzen des erweiterten Kalenders (**)

Beim Ausfüllen der APPT-Maske werden Sie feststellen, daß der Cursor das Jahresfeld (Yr) nicht länger überspringt. Sie können mit zwei Ziffern ein beliebiges Jahr dieses Jahrhunderts spezifizieren.

Beispiel:

```
Day 01/17/84 Hr: Mn AM #1N !Note
```

Eingabe der Jahresinformation.

```
TUE 01/07/84 12:00 AM #1N
```

Legt einen Termin im Jahr 1984 fest.

Ersatzwert für die Zeitfelder ist Mitternacht.

Wenn Sie das Feld Yr überspringen, wird der Termin, falls möglich, im laufenden Jahr fällig. Wenn das gegebene Datum im laufenden Jahr nicht existiert, sucht der HP-75 in den folgenden Jahren nach der gewünschten Übereinstimmung.

Mit [SHIFT] [APPT] können Sie das Jahr der Fälligkeit eines Termins anzeigen:

```
Yr=1984
```

[SHIFT] [APPT] zeigt das Jahr des derzeitigen Termins und sein Wiederholungsintervall (wenn er vom Typ R oder A ist) an.

Die Einstellung EXTD erlaubt Ihnen, Termine in der Vergangenheit zu planen. Geben Sie beispielsweise Ihr Geburtsdatum ein, um den Wochentag Ihrer Geburt zu finden:

Beispiel:

```
Day 04/03/50 Hr:Mn AM #1N !Note
```

Eingabe eines Datums im Jahr 1950.

```
MON 04/03/50 12:00 AM #1N
                      APPT
```

Der Wochentag (Montag) wurde berechnet. Der Termin wird sofort fällig, da 1950 entschieden in der Vergangenheit liegt.

Wenn Sie zwei Sterne (**) in das Feld Yr eingeben, können Sie einen Termin außerhalb des laufenden Jahrhunderts planen.

Beispiel:

```
Day 05/14/** Hr:Mn AM #1N !Note
```

Mit einem Doppelstern im Feld Yr haben Sie Zugriff auf 100 Jahrhunderte.

```
Year? YYYY
```

Der HP-75 antwortet mit der Maske YEAR?: In welchem Jahr soll der Termin fällig werden?

Geben Sie vier Ziffern (0000 bis 9999) ein und drücken Sie [RTN].

Beispiel:

```
Year? 2001
```

Eingabe eines Termins für das Jahr 2001.

```
MON 05/14/01 12:00 AM #1N
```

Der Wochentag und Ersatzwerte werden eingesetzt.

Zeigen Sie mit [SHIFT] [APPT] das Jahr des Termins nochmals an:

```
Yr=2001
```

Der Beispielstermin wird im Jahr 2001 fällig.

Suche nach Datum/Tag

Wenn Sie im erweiterten Kalender ein Datum oder einen Wochentag suchen wollen, geben Sie die gewünschten Werte ein, und lassen Sie den HP-75 den Rest berechnen.

Beispiele:

```
Day 12/25/83 Hr:Mn AM #1N !Note
```

Gesucht: der Wochentag von Weihnachten 1983.

```
SUN 12/25/83 12:00 AM #1N
```

Weihnachten 1983 wird ein Sonntag sein.

```
Day 1 /7 /** Hr:Mn AM #1N !Note
```

Eingabe des Tages, an dem Galilei die drei Monde des Jupiter zuerst entdeckt hat.

```
Year? YYYY
```

Der HP-75 fragt nach dem Jahr.

```
Year? 1610
```

Das Jahr war 1610.

```
THU 01/07/** 12:00 AM #1N
                      APPT
```

Galilei entdeckte die Jupitermonde an einem Donnerstag.

```
tue 11/Dy/84 Hr:Mn AM #1N !Note
```

Welches Datum hat der erste Dienstag im November 1984?

```
TUE 11/06/84 12:00 AM #1N
```

Der erste Dienstag ist der 6. November.

```
sun 2/29/Yr Hr:Mn AM #1N !Note
```

In welchem Jahr fällt der 29. Februar das nächste Mal auf einen Sonntag?

```
SUN 02/29/04 12:00 AM #1N
```

Das nächste solche Schaltjahr ist 2004. (Bestätigen Sie dies mit [SHIFT] [APPT].)

Der HP-75 benutzt für alle Berechnungen im APPT- und im TIME-Modus den Gregorianischen Kalender, der in einigen europäischen Ländern im Jahr 1582 und in Großbritannien und den amerikanischen Kolonien im Jahr 1752 übernommen wurde. Deshalb stimmen Daten vor 1582 nicht mit dem damals benutzten Kalender überein.

Führen Sie im Time-Modus `stats` [RTN] aus, tasten Sie das Wort `year` über die Zeichen `EXTD` und drücken Sie [RTN], wenn Sie zum normalen Jahreskalender zurückkehren wollen. Beim Überprüfen Ihres `appt`-File werden Sie feststellen, daß das Datum der im erweiterten Kalender eingerichteten Termine dabei erhalten bleibt.

Kopieren von Terminen auf und von Massenspeicher (COPY)

Der `appt`-File kann auf und von Massenspeichereinheiten kopiert werden. Drücken Sie [EDIT], wenn Sie den `appt`-File auf Magnetkarte(n) kopieren wollen, und geben Sie ein:

```
>copy appt to card
```

```
Copy to card: Align & [RTN]
```

Der HP-75 führt Sie durch diesen Vorgang. Sobald die Eingabeaufforderung und der Cursor wieder erscheinen, ist die Operation beendet.

Geben Sie die folgende Sequenz ein, wenn Sie Termine von Magnetkarten in den Speicher kopieren wollen:

```
>copy card to appt
```

```
Copy from card: Align & [RTN]
```

Kopiert zuvor aufgezeichnete Termine in den `appt`-File im Speicher.

Diese Operation erzeugt einen `appt`-File, falls er nicht schon existiert.

Der HP-75 mischt die Termine von der Magnetkarte in den derzeitigen `appt`-File ein, so daß alle Termine chronologisch angeordnet sind. Wenn ein Paar identischer Termine existiert, – identisch in jeder Beziehung – , ignoriert der HP-75 den Termin von der Magnetkarte beim Mischen der beiden Files. Siehe auch Abschnitt 9, Seite 135, «Kopieren von Files auf und von Massenspeicher».

Abschnitt 8

Kartenleser-Operationen

Inhalt

Einführung

Mit dem Kartenleser des HP-75 können Sie Textfiles, BASIC-Files, Tastenfiles, LEX-Files und Austauschfiles zwischen dem Speicher und Magnetkarten übertragen. Ein File kann in beide Richtungen übertragen werden:

Speicher* ⟶ *Magnetkarte ***Magnetkarte* ⟶ *Speicher***

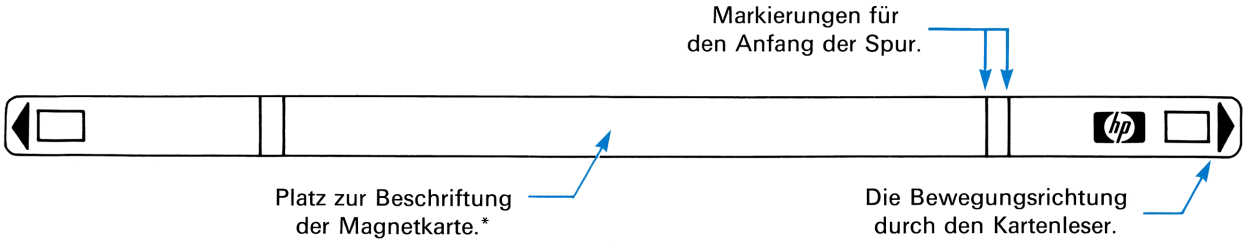
Unabhängig von der Übertragungsrichtung resultiert ein Filetransfer immer in zwei Files: eine Ausgabe des Files steht im Speicher, und eine Ausgabe befindet sich auf der Magnetkarte.

HP-75 Magnetkarten

Jede Magnetkarte besitzt zwei Daten*spuren*, die die folgende Information aufzeichnen:

- Den Katalogeintrag des auf der Spur aufgezeichneten Files.
- Die Gesamtzahl von Spuren des Files.
- Die Kennzahl *dieser* Spur, eine Zahl zwischen 1 und der Gesamtzahl von Spuren des Files.
- Das Paßwort, falls vorhanden. (Paßworte können nicht angezeigt werden.)
- Der Schreibschutz-Status der Spur; d.h. die Information, ob die Spur gegen Überschreiben geschützt ist.
- Bis zu 650 Bytes des Fileinhalts. Eine Spur kann nur Information aus einem einzigen File aufnehmen. Eine *Magnetkarte* kann Information aus einem oder zwei Files aufnehmen.

Beim Durchziehen jeder Spur einer Magnetkarte durch den Kartenleser muß immer die bedruckte Seite der Magnetkarte nach oben weisen. Die Reihenfolge der Spuren spielt keine Rolle.



*siehe Anhang B.

Wichtig: Halten Sie Magnetkarten sauber und frei von Öl, Fett und Schmutz, und fassen Sie sie nur an ihren Kanten an. Schmutz und Fingerabdrücke beeinträchtigen die Zuverlässigkeit des Kartenlesers, erzeugen Warnmeldungen und vermindern die Lebensdauer der Magnetkarten. Sie können die Magnetkarten mit Isopropyl-Alkohol und einem weichen Tuch reinigen. Halten Sie die Magnetkarten von starken Magnetfeldern (wie von Dauermagneten, Starkstromleitungen, Netztransformatoren und Entmagnetisiergeräten) fern; Magnetismus kann die Magnetkarten dauerhaft beschädigen. Siehe auch Anhang B.

Benutzung des Kartenlesers

Der HP-75 zeigt während einem Einlesevorgang eine Vielzahl von Meldungen an. Wenn eine Meldung eine Anzeigezeile überschreitet, bleibt die erste Zeile der Meldung entsprechend der derzeitigen Verzögerungsrate sichtbar. Drücken Sie [SHIFT] [FET] und halten Sie die Taste [FET] gedrückt, wenn Sie die erste Zeile einer Magnetkartenlesemeldung nochmals sehen wollen.

Für alle Magnetkartenleseoperationen sind die folgenden Schritte erforderlich:
Geben Sie im EDIT-Modus bei angezeigter Text- oder BASIC- Eingabeaufforderung (: oder >) einen Kartenleserbefehl (zum Beispiel `COPY TO CARD`) ein. Sie können Groß- und Kleinbuchstaben in beliebiger Kombination verwenden.

Starten Sie die Operation mit [RTN]. Der HP-75 antwortet mit der zugehörigen Meldung und wartet.

Beispiel:

```
2 track(s) needed
```

Eine solche Meldung erscheint immer, wenn Sie einen File auf Magnetkarten kopieren wollen. Die Meldung wird entsprechend der derzeitigen Verzögerungsrate angezeigt und kann mit [SHIFT] [FET] kurzzeitig zurückgerufen werden.

```
Copy to card: Align&ERTN
```

Der HP-75 wartet danach auf Ihre Reaktion. Sie können mit [ATTN] die Operation abbrechen.

Schieben Sie die Magnetkarte in der Richtung der gewünschten Spur so in den Leseschlitz ein, daß die rechte Markierung gerade neben dem Schlitz steht; die Magnetkarte sollte soweit auf der anderen Seite herausragen, daß der Pfeil und das Rechteck sichtbar werden. Drücken Sie dann ein zweites Mal [RTN]. Der HP-75 antwortet:

```
Pull card...
```

Ziehen Sie die Magnetkarte durch den Leseschlitz. Sie haben 5 Sekunden Zeit, den Einlesevorgang zu beginnen. Wenn Sie länger zögern, erzeugt der HP-75 ein Tonsignal und zeigt Warnung 23 – `bad read/write` – an, und fordert Sie auf, die Operation zu wiederholen. Wenn Sie sich entschließen, die Magnetkarte *nicht* durchzuziehen, warten Sie, bis Sie der HP-75 nach etwa 5 Sekunden wieder zum Einführen der Magnetkarte auffordert (`Align & ERTN`), und drücken Sie stattdessen [ATTN].

Nach dem Durchziehen der Magnetkarte durch den Leser sind verschiedene Reaktionen des HP-75 möglich:

- Wenn Sie einen File vom Speicher auf eine Magnetkarte übertragen, fordert Sie der HP-75 zu einem zweiten Durchgang der gleichen Spur auf:

```
Verify Card: Align&ERTN
```

Bei diesem Durchgang wird die übertragene Information auf ihre Fehlerfreiheit geprüft. Wenn die übertragene Information nicht mit der ursprünglichen Information identisch ist, zeigt der HP- 75 die Warnung 21 – `verify failed` – an, und verlangt zwei weitere Durchgänge der gleichen Spur, einen zum erneuten Überschreiben und einen zum Prüflesen. (Wenn diese Magnetkarte dabei immer noch nicht ordnungsgemäß beschrieben werden kann, sollten Sie die Magnetkarte reinigen oder eine neue Magnetkarte verwenden.)

- Wenn der File mehr als eine Spur umfaßt, meldet Ihnen der HP-75, wenn die erste Spur aufgezeichnet ist:

`Track 1 done; insert track 2`	Diese Meldung wird mit der derzeitigen Verzögerungszeit angezeigt, danach...
`Copy to card: Align & [RTN]`	Fertig zum Umdrehen der Magnetkarte oder Einführen einer neuen Magnetkarte.

- Wenn Sie eine Magnetkarte zu schnell oder zu langsam durchziehen, zeigt der HP-75 die Warnung 24 – `pulled too fast` – oder 25 – `pulled too slow` – an, und fordert einen weiteren Durchgang an. Wenn Sie eine Magnetkarte *sehr* langsam durchziehen, kann der HP-75 so reagieren, als wenn gar keine Magnetkarte durchgezogen worden wäre.
- Der HP-75 wiederholt seine Aufforderungen für soviele Durchgänge, wie erforderlich sind. Wenn die Operation abgeschlossen ist, erscheinen der Cursor und die Eingabeaufforderung des EDIT-Modus wieder in der Anzeige.

Der HP-75 verfügt über fünf Kartenleserbefehle. Dies sind, in ihrer einfachsten Form:

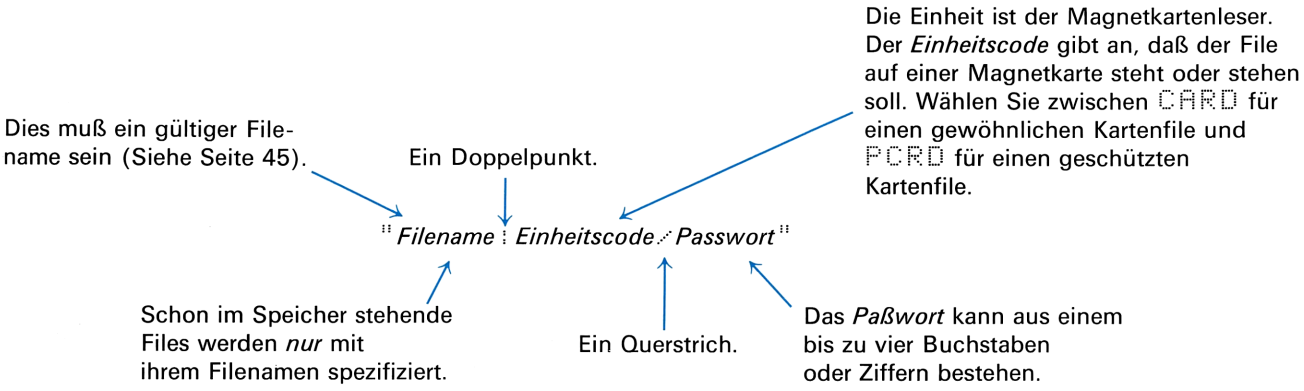
`CAT CARD`	Anzeige des Katalogeintrags eines Kartenfiles.
`COPY TO CARD`	Kopieren des momentanen Files im Speicher auf Magnetkarten.
`COPY CARD TO '`*Filename*`'`	Einlesen eines Kartenfiles in den Speicher.
`PROTECT`	Schützen einer einzelnen Spur gegen Überschreiben.
`UNPROTECT`	Entfernen des Schreibschutzes.

Nach einer Kartenleseroperation zeigt der Filepointer auf die gleiche Zeile des gleichen BASIC- oder Textfiles wie *vor* der Operation. Wenn Sie einen neu eingelesenen File editieren wollen, müssen Sie also zuerst einen `EDIT`-Befehl ausführen.

> **Programmierhinweis:** Alle fünf Kartenleserbefehle sind programmierbar. Während der Programmausführung initiiert ein Kartenleserbefehl die entsprechende Operation. Nach der Ausführung dieser Operation wird die Programmausführung mit der nächsten Anweisung im Programm fortgesetzt. (Kopieren des derzeit in Ausführung befindlichen Files stellt eine Ausnahme dar – in diesem Fall wird der den Programmvariablen zugewiesene Speicherplatz freigegeben, der File wird kopiert und die Programmausführung angehalten.)

Spezifizieren von Magnetkartenfiles

Sie haben verschiedene Möglichkeiten, einen bestimmten Magnetkartenfile zu *spezifizieren,* oder zu benennen. Am einfachsten ist es, in einem Befehl das Schlüsselwort `CARD` (ohne Anführungszeichen) zu spezifizieren, unabhängig davon, welcher File momentan auf der Magnetkarte ist. Ein Karten*filespezifikator* ist ein Zeichenstring in Anführungszeichen, der einen *bestimmten* Kartenfile spezifiziert. Ein Kartenfilespezifikator enthält einen *Filenamen*, einen *Einheitscode* und optional ein *Paßwort*.



Der gesamte Filespezifikator muß in Anführungszeichen (' ' oder " ") stehen. Kleinbuchstaben im Filespezifikator einschließlich des Paßworts werden als Großbuchstaben interpretiert.

Die Spezifikation von PCRD bewirkt, daß der resultierende geschützte Magnetkartenfile wieder wie jeder andere File in den Speicher kopiert werden kann, daß er aber, wenn er wieder im Speicher steht, auf keine Weise untersucht oder editiert werden kann. Nur BASIC-Files oder Programme können geschützt werden. Geschützte Files werden im Systemkatalog mit PB kenntlich gemacht; Sie können die Befehle RUN, PURGE und CALL, nicht aber die Befehle LIST, PLIST, EDIT, COPY oder RENAME mit geschützten Files durchführen. Mit PCRD können Sie verhindern, daß Benutzer Fertigprogramme betrachten, ändern oder duplizieren können.

Wenn Sie beim Kopieren eines Files auf Magnetkarte ein Paßwort spezifizieren, wird der File mit diesem Paßwort gezeichnet. Wenn Sie einen File mit Paßwort von einer Magnetkarte einlesen, müssen Sie das Paßwort im Filespezifikator mit angeben; andernfalls erhalten Sie Fehler 66 – invalid password –, und die Kopieroperation wird abgebrochen. Wenn Sie ein Paßwort spezifizieren, können Sie die Benutzung des entsprechenden Files auf Besitzer des Paßwortes beschränken.

Beispiele:

'anruf:card'	Spezifiziert den Kartenfile ANRUF. Beachten Sie, daß 'anruf' allein den File ANRUF im *Speicher* spezifizieren würde.
'bywcal:pcrd'	Spezifiziert den Kartenfile bywcal, einen geschützten Kartenfile.
'Monmtng:card/11'	Spezifiziert den Kartenfile MONMTNG, der mit dem Paßwort 11 gesichert ist.
'A10352:pcrd/rot'	Spezifiziert den geschützten BASIC-Kartenfile A10352, der mit dem Paßwort ROT gesichert ist. Dies ist die höchste Sicherheitsstufe.

Beachten Sie, daß Sie auch einen Einheitscode (CARD oder PCRD) angeben müssen, wenn Sie einen File mit einem Passwort spezifizieren wollen, zum Beispiel: 'MONMTNG:CARD/11'.

Hinweis für den fortgeschrittenen Benutzer: Sie können passende Stringvariablen oder Ausdrücke als Parameter in Kartenleserbefehlen verwenden. Wenn der Stringvariablen A$ zum Beispiel 'raan:card/ry' zugeordnet wurde, kann A$ als Filespezifikator des Kartenfiles RAAN mit dem Paßwort RY verwendet werden. (Das ohne Anführungszeichen stehende Schlüsselwort CARD kann jedoch nicht durch eine Stringvariable spezifiziert werden.) Filespezifikatoren können auch mit Leerstellen aufgefüllt werden; 'Ohms:card revision A' spezifiziert zum Beispiel den Kartenfile OHMS.

Der Katalogeintrag einer Magnetkarte (CAT CARD)

Der Befehl CAT CARD ermöglicht Ihnen, den Katalogeintrag eines beliebigen Magnetkartenfiles zu überprüfen.

```
CAT CARD
```

Geben Sie cat card RTN ein, um den Katalogeintrag einer vorbespielten Magnetkarte – zum Beispiel dem Ihrem HP-75 beigelegten Programm FINANZ – anzuzeigen.

```
>cat card
```

```
Catalog card: Align & ERTN
```

Die Antwort des HP-75.

Schieben Sie eine beliebige Spur des Programms FINANZ in den Kartenleser (wir benutzen die erste Spur) und drücken Sie RTN:

```
Pull card...
```

Der HP-75 fordert Sie auf, die Magnetkarte durch den Schlitz des Kartenlesers zu ziehen.

Nach gleichmäßiger, glatter Durchführung erhalten Sie die folgende Anzeige:

```
FINANZ    (track  1 of  4)
```

Anzeige der Kennummer der momentanen Spur und der Spurenzahl des Kartenfiles.

Nach der Verzögerungszeit erscheint der Katalogeintrag:

```
FINANZ    B    2248 13:52 07/13/82
```

Zeigen Sie mit SHIFT FET kurzzeitig die erste Zeile an. Nach Drücken jeder anderen Taste – außer nach ←, →, SHIFT oder CTL – erscheinen der Cursor und die Eingabeaufforderung wieder.

Beachten Sie, daß Sie mit einem CAT CARD-Vorgang den Katalogeintrag einer Spur nur *anzeigen* können. Der File selber wird mit CAT CARD nicht übertragen. Wenn ein Programmfile ein geschützter Kartenfile ist (d.h. wenn er mit der Spezifikation PCRD auf die Magnetkarte geschrieben wurde), wird der Filetyp mit PB gekennzeichnet. Paßworte werden nie angezeigt und müssen für den Befehl CAT CARD nicht eingegeben werden.

Sie können anstelle von CAT CARD auch CAT ':CARD' oder CAT ':PCRD' verwenden. Die drei Befehle sind in ihrer Wirkung identisch. CAT CARD Befehle sind nützlich, um den Katalogeintrag von nicht beschrifteten Magnetkarten zu untersuchen.

Wenn Sie den Katalogeintrag einer noch unbenutzten Spur abrufen, erhalten Sie folgende zweizeilige Meldung:

```
blank    <track  1 of 1>
```

```
blank     B     0 00:00 01/01/00
```

Der Katalogeintrag zeigt, daß diese Spur zuvor noch nicht benutzt wurde.

Beschreiben einer Magnetkarte (COPY TO CARD)

Die allgemeinste Form des Befehls COPY hat folgendes Format:

```
COPY Quelle TO Ziel
```

wobei ein Kartenfile sowohl als *Quelle* wie auch als *Ziel* auftreten kann. Benutzen Sie eine der folgenden Formen, wenn Sie einen File auf Magnetkarten kopieren wollen:

```
COPY ['Filename'      TO CARD
      KEYS                'Filespezifikator'
      APPT                ':CARD'
                          ':PCRD'
```

Beispiele:

```
>copy to card
```

Die einfachste Form des Befehls COPY kopiert den momentanen BASIC- oder Textfile. Alle Information, einschließlich Name und Zeit des Quellfiles wird kopiert. Der Kartenfile wird nicht gesichert.

```
>copy to ':pcrd'
```

Der Quellfile muß ein BASIC-File sein; der Ziel-Kartenfile ist gegen Listen und Editieren geschützt.

```
>copy to 'konten: pcrd'
```

Der momentane File wird als privater Kartenfile kopiert. Wenn der Zielfilename mit dem Quellfilenamen übereinstimmt, werden Datum und Uhrzeit übernommen; andernfalls erhält der Kartenfile das aktuelle Datum und Uhrzeit.

```
>copy 'thema' to card
```

Der spezifizierte Quellfile (THEMA) wird kopiert; der Filepointer bleibt fixiert. Datum und Zeit stimmen überein. Keine Sicherung.

```
>copy 'thema' to 'thema1:card'
```

Neues Datum, neue Zeit. Keine Sicherung.

```
>copy 'thema' to 'thema:card/h'
```

Durch das Paßwort H wird der Zugriff beschränkt. Datum und Zeit stimmen überein.

```
>copy appt to 'Jan:card/wry'
```

Kopiert den appt-File in einen File mit dem Namen JAN und dem Paßwort WRY. Neue Zeit, neues Datum.

```
>copy keys to card
```

Kopiert den File keys. Keine Sicherung. Neue Zeit, neues Datum.

Diese Beispiele zeigen, daß verschiedene Filenamen für Quelle und Ziel verschiedene Zeiten und ein anderes Datum ergeben. Nur BASIC-Files – im Gegensatz zu Text-, Termin-, Tasten- und anderen Files – können geschützt werden. Wenn BRIEF zum Beispiel ein Textfile ist, erzeugt die Eingabe `copy 'brief' to ':pcrd'` RTN Warnung 68 – `wrong filetype`. Der Kopiervorgang kann fortgesetzt werden, aber der kopierte Kartenfile ist nicht geschützt. Sie können Text-, Termin-, Tasten- und andere Files mit *Paßwörtern* sichern, wenn Sie den Kopierbefehl mit vollständigem Filespezifikator eingeben. Erinnern Sie sich, daß zur Spezifikation von Files im *Speicher* nie Einheitscodes oder Paßwörter benutzt werden.

Wenn ein File aus mehr als 650 Bytes besteht, wird mehr als eine Spur zu dessen Aufzeichnung benötigt. Der HP-75 berechnet die notwendige Anzahl Spuren zu Beginn einer COPY-Operation und beginnt den Kopiervorgang mit Spur 1. Nachdem Sie eine COPY TO CARD-Operation abgeschlossen haben, können Sie mit dem Befehl CAT CARD den Katalogeintrag jeder Spur abrufen.

Einlesen einer Magnetkarte (COPY CARD)

Benutzen Sie eine der folgenden Formen des COPY Befehls, um einen Kartenfile in den Speicher einzulesen:

```
COPY  CARD          TO  'Filename'
      ':CARD'           APPT
      ':PCRD'           KEYS
      'Filespezifikator'
```

Die *Quelle* ist der zu kopierende Kartenfile, und das *Ziel* ist der neue File im Speicher, so wie er im Systemkatalog erscheint. Sie benötigen nur dann einen vollständigen Filespezifikator, wenn der File mit einem Paßwort gesichert ist. Bei anderen Kartenfiles können die Spezifikationen `CARD` oder `':CARD'` gleichwertig nebeneinander für Einlesebefehle benutzt werden.

Beispiele:

```
>copy card to 'eh'
```

Erzeugt den File `EH` im Speicher und kopiert den Inhalt des Kartenfiles auf diesen (Annahme: kein Paßwort). Falls der Kartenfile ebenfalls EH heißt, werden Zeit und Datum vom Originalfile übernommen; andernfalls gelten momentane Zeit und Datum.

```
>copy ':card' to 'eh'
```

Gleiche Funktion wie oben. Es darf jedoch nur ein File mit dem Namen `EH` im Speicher existieren.

```
>copy 'konten:card' to 'a'
```

Stellt fest, ob die eingeführte Spur zum spezifizierten Kartenfile (`Konten`) gehört. Wenn ja, wird ein File `A` mit neuer Zeit und neuem Datum im Speicher erzeugt; andernfalls erscheint Warnung 26 – `wrong name` – und der HP-75 erwartet eine andere Magnetkarte. Ist `KONTEN` auf der Magnetkarte ein privater File, dann ist `A` im Speicher ebenfalls ein privater File.

```
>copy 'thema:card/h' to 'test'
```

Prüft Filenamen und Paßwort des Kartenfiles. Falls diese korrekt sind, wird der File `TEST` mit neuer Zeit und neuem Datum im Speicher erzeugt.

```
>copy 'Jan:card/wry' to appt
```

Gliedert die Termine auf der Magnetkarte in den momentanen Terminfile ein.

```
>copy card to keys
```

Erzeugt einen neuen File `keys`, der das Tastenfeld umdefiniert, sobald er eingelesen ist. (Es darf nur ein File `keys` im Speicher vorhanden sein.) Siehe auch Abschnitt 10.

```
>copy ':pcrd' to 'nurlauf'
```

Erzeugt den File `NURLAUF` im Speicher und kopiert den Kartenfile dorthin.

Das letzte Beispiel illustriert den Gebrauch von `':PCRD'` bei der Spezifikation einzulesender Kartenfiles. Diese Form des Befehls `COPY` ergibt einen privaten File im Speicher, auch wenn der Kartenfile nicht privat ist. Der mit `':PCRD'` spezifizierte Kartenfile sollte ein BASIC-File sein, oder Sie erhalten Warnung 68 – `wrong file type`; der Kopiervorgang kann jedoch fortgesetzt werden ohne dabei einen geschützten File zu erzeugen.

Die Angabe eines vom Quellfilenamen verschiedenen Zielfilenamens erzeugt ein neues Datum und eine neue Uhrzeit im Systemkatalog.

Wennn bei einem Einlesevorgang eine Spur durch den Magnetkartenleser gezogen wird, stellt der HP-75 fest, ob genügend Speicherplatz für den gesamten File vorhanden ist; ist dies nicht der Fall, wird Fehler 16 – `not enough memory` – angezeigt und der Kopiervorgang abgebrochen.

Alle Spuren des Kartenfiles müssen eingelesen werden, damit der File im Systemkatalog eingetragen wird. Das bedeutet, daß der Kartenfile ganz aus dem Speicher verschwindet, wenn ein Einlesevorgang unterbrochen wird. Wenn Sie den gleichen File vom Speicher auf zwei Magnetkartensätze kopieren, können danach Spuren beider Magnetkartensätze gemischt wieder eingelesen werden. Beachten Sie jedoch, daß die beiden Magnetkartensätze *identische* Kopien des selben Files mit identischen Katalogeinträgen sein müssen; andernfalls erhalten Sie die Warnung 20 – not this file –, wenn Sie die Spuren bei einem Einlesevorgang mischen.

Schützen von Kartenfiles (PROTECT, UNPROTECT)

Der Magnetkartenleserbefehl PROTECT ermöglicht Ihnen, Kartenfiles gegen Löschen (zum Beispiel durch Überschreiben) zu schützen.

```
PROTECT
```

Sie müssen für jede zu schützende *Spur* den Befehl PROTECT ausführen:

```
>protect
```

```
Protect card:  Align & [RTN]
```

Löst die Operation aus. Die nächste durch den Magnetkartenleser gezogene Spur wird gegen Überschreiben geschützt.

Nachdem die Spur durch den Magnetkartenleser gezogen wurde, erscheinen der Cursor und die Eingabeaufforderung wieder; die Spur ist dann gegen nachfolgendes Beschreiben geschützt. PROTECT kann jederzeit ausgeführt werden, im Normalfall werden Sie den Befehl jedoch direkt an den Kopiervorgang auf die Magnetkarte anschließen.

Sie müssen für jede zu schützende Spur eine eigene PROTECT Operation durchführen. Mit dem folgenden Programm können Sie nacheinander eine Reihe von Spuren gegen Überschreiben schützen:

```
>10 PROTECT @ GOTO 10
```

Drücken Sie [RUN] oder geben Sie run ein, um dieses Programm zu starten, wenn mehrere Spuren gegen Überschreiben geschützt werden müssen. Das Programm führt den Befehl PROTECT so lange hintereinander aus, bis Sie das Programm mit [ATTN] unterbrechen.

Mit dem Befehl UNPROTECT können Sie die Schreibschutzmarkierung wieder vom Spurkopf entfernen.

```
>unprotect
```

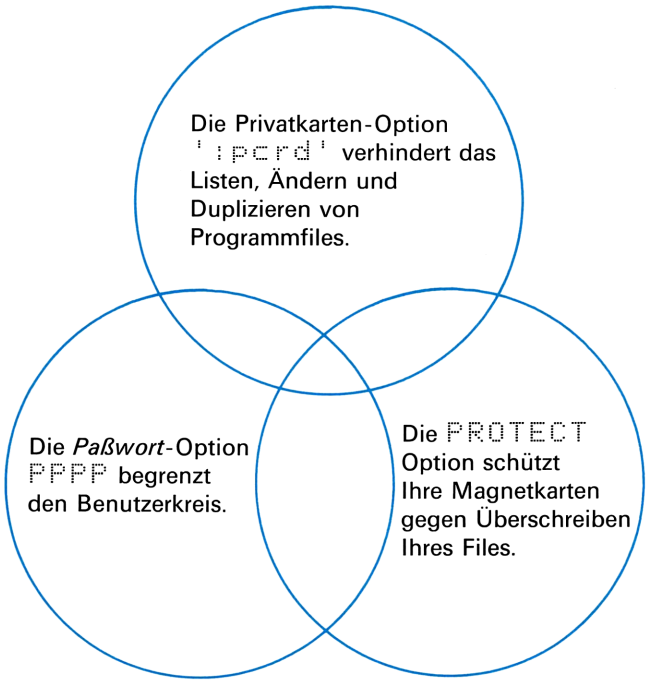
```
Unprotect card:  Align & [RTN]
```

Der Schreibschutz der nächsten durch den Kartenleser gezogenen Magnetkarte wird entfernt.

Nach einer UNPROTECT-Operation ist eine Spur wieder zur Datenaufnahme verfügbar. Beachten Sie, daß der HP-75 über keine explizite Anweisung zum Löschen von Magnetkarten verfügt: eine bespielte Spur wird gelöscht, wenn Sie sie mit einem neuen File überschreiben.

Die drei P's der Sicherung

Sie haben drei Möglichkeiten zur Sicherung von Kartenfiles kennengelernt:



Diese Optionen können einzeln angewandt oder kombiniert werden und ergeben eine Vielzahl von Schutzmöglichkeiten für Ihre Files. Die Schutzfunktionen des HP-75 sind jedoch eher zur Verhinderung ungewollter Änderung oder Duplizierung wichtiger Files gedacht, als dazu, absolute Sicherheit zu garantieren.

Abschnitt 9

HP-IL Operationen

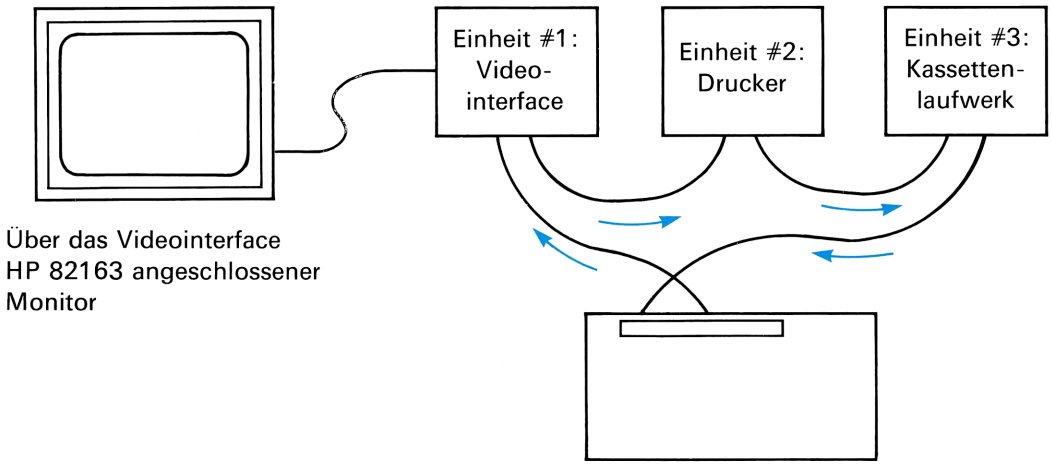
Inhalt

Einführung

Als alleinstehender Computer benutzt der HP-75 das Tastenfeld als *Eingabeeinheit*, das Anzeigefenster und den Tonsignalgeber als *Ausgabeeinheiten*, und den Kartenleser sowohl als Eingabe- wie auch als Ausgabeeinheit. Mit Hilfe der Hewlett-Packard Interface Loop (Interfaceschleife) des HP-75 können Sie eine Vielzahl weiterer externer I/O-Einheiten mit dem Computer steuern; bis zu 30 Peripheriegeräte einschließlich Bildschirminterfaces, Druckern und Kassettenlaufwerken sind möglich.

In diesem Abschnitt wird die Arbeitsweise der HP-IL anhand des HP-75 als Steuereinheit, des Videointerfaces HP82163 als Anzeigeeinheit, des Thermodruckers HP82162A als Druckereinheit und des Digital-Kassettenlaufwerks HP82161A als Massenspeichereinheit illustriert. Der HP-75 und die HP-IL Peripheriegeräte sind in einer einzigen Schleife, einem Kommunikationskreis, zusammengeschlossen.

Ein HP-IL System für Anzeige, Ausdruck und Speicherung von Information.



Anweisungen und Daten in der Hewlett-Packard Interface Loop werden im Computer erzeugt und wandern von einer Einheit zur nächsten in der Schleife, bis diese Information den HP-75 wieder erreicht. Wenn die Information nicht für eine spezielle Einheit bestimmt ist, wird sie einfach jeweils an die nächste Einheit weitergegeben. Sobald die Information die richtige Einheit erreicht, reagiert diese den Anweisungen entsprechend und gibt die Information dann weiter. Der Aufbau eines HP-IL Systems besteht aus zwei Schritten:

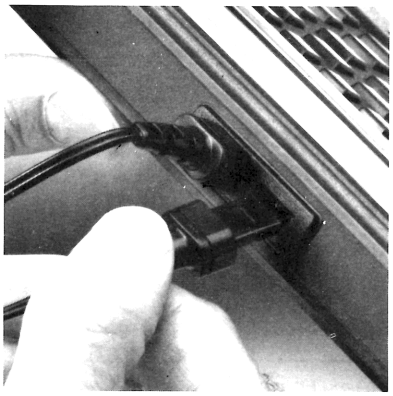
1. Die Einheiten in der Schleife müssen physikalisch verbunden werden.
2. Sie müssen den Einheiten individuelle Einheitscodes zuteilen, damit Sie die Geräte für spezifische Funktionen einzeln ansprechen können.

Anschluß der Hewlett-Packard Interface Loop

Anschluß einer oder mehrerer Einheiten an den HP-75:

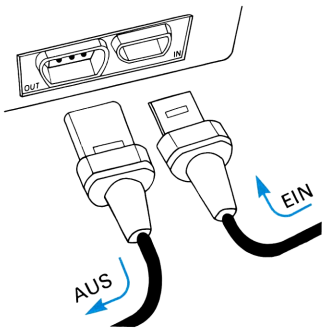
1. Bereiten Sie jede Einheit entsprechend der Installationsanleitung im Benutzerhandbuch des Peripheriegerätes auf den Anschluß vor.
2. Schalten Sie alle HP-IL Einheiten ein, nachdem Sie sich versichert haben, daß Sie die richtigen Versorgungsspannungen gewählt haben. Der HP-75 kann dabei ein- oder ausgeschaltet sein.
3. Schließen Sie die HP-IL Kabel an. Zwei 1 Meter lange HP-IL Kabel wurden Ihrem HP-75 beigelegt; jede Einheit besitzt ein passendes Kabel.

Alle Anschlüsse sind gekennzeichnet, um die richtige Orientierung zu gewährleisten.



4. Sie können die Einheiten in beliebiger Reihenfolge anschließen, aber die endgültige Konfiguration ist wichtig. Die an die Buchse OUT des HP-75 angeschlossene Einheit wird Einheit Nr.1, die an die Buchse OUT von Einheit Nr.1 angeschlossene Einheit wird Einheit Nr.2 und so fort. Die Einheit, die an die Buchse IN des HP-75 angeschlossen ist, ist die letzte Einheit in der Schleife.

Die Form der Buchsen zeigt die Flußrichtung in der Schleife an.



Wenn Sie die Peripheriegeräte weiter voneinander entfernt aufstellen wollen, als deren Kabel erlauben, fügen Sie einfach an den entsprechenden Stellen weitere HP-IL Kabel in die Schleife ein. Der maximale Abstand zwischen zwei Einheiten ist 10 Meter, wenn Sie Standardkabel verwenden.
Alle HP-IL Befehle sind programmierbar; der Einfachheit halber werden sie in den Beispielen vom Tastenfeld aus ausgeführt.

Hinweis: Um unerwartete Unterbrechungen des HP-IL Betriebs zu vermeiden, sollten Sie die Peripheriegeräte erst nach dem Ausschalten des HP-75 ebenfalls ausschalten. Schalten Sie alle Peripheriegeräte vor dem HP-75 ein.

Zuweisen von Einheitscodes (ASSIGN IO)

Nachdem die Einheiten mit ihrer Stromversorgung und in der Interfaceschleife verbunden sind, können Sie ihnen Einheitscodes zuweisen, um sie individuell ansprechen zu können. Schalten Sie mit [EDIT] in den EDIT-Modus und geben Sie `assignio` [RTN] ein:

```
>assignio
```

Beide Eingabeaufforderungen (`>` oder `:`) sind möglich.

```
3 Device(s) on loop
```

Diese Meldung enthält die Anzahl Einheiten, die in der Schleife verbunden sind. Sie wird entsprechend der momentanen Verzögerungsrate angezeigt.

Diese Meldung und andere HP-IL Meldungen werden kurzzeitig nochmals angezeigt, wenn Sie [SHIFT] [FET] drücken. Der HP-75 fragt nach dem Einheitscode der ersten Einheit:

```
Device # 1=':
```

Einheit 1 ist die an die Buchse OUT des HP-75 angeschlossene Einheit, in unserem Beispiel das Videointerface.

Hinweis: Wenn der HP-75 nicht wie angegeben reagiert, und der Befehl ASSIGN IO «starr» in der Anzeige stehen bleibt, können die HP-IL Informationen nicht einwandfrei in der Schleife wandern. Löschen Sie die Anzeige mit [ATTN], vergewissern Sie sich, daß alle Einheiten angeschlossen und mit Strom versorgt sind, und geben Sie den Befehl nochmals ein.

Einheitscodes können aus einem oder zwei Buchstaben, einem Buchstaben und einer Zahl oder einer Zahl und einem Buchstaben bestehen. Beispiele für erlaubte Einheitscodes sind `t`, `tv`, `t1` und `1t`. (Ein Leerzeichen als letztes Zeichen eines Einheitscodes wird ignoriert; ein Leerzeichen als erstes Zeichen ist unzulässig.) Die Buchstaben eines Einheitscodes werden intern in Großbuchstaben umgewandelt.

Geben Sie den ersten Einheitscode mit [RTN] ein:

```
Device # 1=':tv'
```

Eingabe von `TV` als ersten Einheitscode.

```
Device # 2=':
```

Sobald Sie [RTN] drücken, verlangt der HP-75 die Eingabe des nächsten Einheitscodes.*

*Sie können eine beliebige Wagenrücklauf/Zeilenvorschubtaste drücken, um die Eingabe der Einheitscodes abzuschließen (Seite 43).

Setzen Sie die Zuteilung der Einheitscodes fort, bis alle Einheiten in der Schleife benannt sind; danach erscheinen der Cursor und die Eingabeaufforderung wieder. Die Einheitscodes bleiben solange im Speicher erhalten, bis ein weiterer ASSIGN IO Befehl ausgeführt wird.

Wenn Sie einen unzulässigen Einheitscode eingeben, meldet der HP-75 einen Fehler und fragt nach einem anderen Namen für die jeweilige Einheit. Wenn Sie für eine Einheit keinen Einheitscode eingeben und RTN drücken, wird der Vorgang beendet, und es sind nur die Einheiten verfügbar, denen Sie einen Einheitscode zugewiesen haben.

Ab diesem Zeitpunkt werden Sie einzelne Geräte in HP-IL Befehlen über die ihnen zugewiesenen Einheitscodes spezifizieren. Einheitscodes müssen in Anführungszeichen (' ' oder " ") eingeschlossen und durch einen Doppelpunkt (:) eingeleitet werden.

Mit einer weiteren Form des Befehls ASSIGN IO können Sie alle Einheitscodes auf einmal zuweisen:

```
ASSIGN IO  ':Einheitscode [,:Einheitscode...]'
```

Den Einheiten wird ihr Einheitscode, ihrer Reihenfolge in der Schleife entsprechend, von Nr.1 bis Nr.30, zugewiesen.

Beispiel:

```
>assignio ':tv,:p,:ca'
```

Mit RTN weisen Sie drei Peripheriegeräten die Einheitscodes TV, P und CA zu.

Die gesamte Einheitsliste steht in Anführungszeichen; Einheitscodes werden durch Doppelpunkte eingeleitet und durch Kommata getrennt. Einheitscodes werden von links nach rechts zugewiesen, TV zum Beispiel wird der Einheit Nr.1 zugewiesen.

> **Hinweis für den fortgeschrittenen Benutzer:** Die Einheitsliste kann durch einen Stringausdruck oder eine Stringvariable spezifiziert werden. Wenn D$ der Ausdruck ':tv,:p,:ca' zugewiesen wurde, bestimmt ASSIGN IO D$ drei Einheitscodes.

Wenn weniger Einheitscodes zugewiesen wurden, als Einheiten vorhanden sind, werden alle Einheitscodes benutzt. Wenn mehr Einheitscodes zugewiesen werden, als Einheiten vorhanden sind, erhalten Sie die Warnung 59 – too many names –, und es werden so viele Einheiten benannt, wie vorhanden sind. Diese Form des Befehls ASSIGN IO ist in Programmen und Tastenumdefinitionen nützlich.

Die meisten HP-IL Operationen erzeugen Fehler 55 – ASSIGN IO needed –, wenn sie ohne vorherige Zuweisung von Einheitscodes ausgeführt werden.

Listen von Einheitszuweisungen (LIST IO)

Prüfen Sie die derzeitigen Zuweisungen der Einheitscodes mit dem Befehl LIST IO.

```
LIST IO
```

Beispiel:

```
>listio
```

Drücken Sie RTN.

```
 3 Device(s) assigned
Device # 1=':TV'
Device # 2=':P '
Device # 3=':CA'
```

Die Anzahl zugewiesener Einheiten wird mit der derzeitigen DELAY Verzögerungsrate angezeigt, und die Einheitscodes werden nacheinander aufgelistet.

Mit ATTN können Sie einen LIST IO Vorgang abbrechen.

Bestimmen von Anzeige- und Druckereinheiten (DISPLAY IS, PRINTER IS)

Verwenden Sie bei der Benutzung Ihrer Anzeige- und Druckereinheiten die den Geräten zugeteilten Einheitscodes.

```
DISPLAY IS ': Einheitscode [, : Einheitscode...]'
```

```
PRINTER IS ': Einheitscode [, : Einheitscode...]'
```

Der Befehl DISPLAY IS bewirkt, daß auf der spezifizierten Peripherieeinheit alle in der Anzeige des HP-75 erscheinenden Informationen protokolliert werden.

Beispiel:

```
>display is ':tv'
```

Wenn Sie RTN drücken, wird die spezifizierte Einheit – hier das Videointerface – als Anzeigeeinheit definiert.

Die Anzeige des HP-75 dient immer als DISPLAY IS Einheit.

Der Befehl PRINTER IS läßt die spezifizierte(n) Einheit(en) auf PRINT-Anweisungen und PLIST-Befehle reagieren.

Beispiel:

```
>printer is ':p'
```

Wenn Sie RTN drücken, wird die spezifizierte Einheit als Druckereinheit deklariert.*

Die Anzeige des HP-75 arbeitet als Druckereinheit, bis mit dem Befehl PRINTER IS eine andere Einheit deklariert wird.

Sie können mit dem Befehl PRINTER IS oder DISPLAY IS mehr als eine Einheit spezifizieren.

Beispiel:

```
>printer is ':p,:tv'
```

Dieser Befehl deklariert beide Geräte als Druckgeräte.

Wenn ein einzelnes Peripheriegerät gleichzeitig als Drucker- und als Anzeigeeinheit deklariert wird, gibt es sowohl Drucker- wie auch Anzeigeinformation aus. Die Deklarationen DISPLAY IS und PRINTER IS machen das spezifizierte Gerät zu einem *Listener* (Empfänger). Schlagen Sie die Listener-Reaktionen auf HP-IL Meldungen im Benutzerhandbuch Ihres Peripheriegerätes nach.

> **Hinweis für den fortgeschrittenen Benutzer:** Wenn eine Anzeige- oder Druckereinheit über einen Buffer verfügt, gibt sie nur Informationen aus, wenn der Buffer gefüllt ist, oder wenn sie vom HP-75 direkt dazu aufgefordert wird. Wenn Sie eine der Wagenrücklauf/Zeilenvorschubtasten (Seite 43) drücken, drucken alle DISPLAY IS Einheiten den Inhalt ihres Buffers aus und springen auf die nächste Zeile. Schlagen Sie Informationen über den Buffer im Benutzerhandbuch Ihres Peripheriegerätes nach.

* Der Schalter MODE und die Taste PRINT des Thermodruckers HP82162A haben keine Wirkung auf den Drucker, solange er mit dem HP-75 verbunden ist.

Die folgenden Befehle heben die Anzeige- und Druckerzuweisungen wieder auf:

```
DISPLAY IS *
DISPLAY IS ''
```

```
PRINTER IS *
PRINTER IS ''
```

Nach Ausführung von DISPLAY IS * werden die folgenden Anzeigeinformationen nur noch auf die Anzeige des HP-75 gelenkt; wenn Sie PRINTER IS * eingeben, wird die nachfolgende Druckerinformation auf die DISPLAY IS Einheiten gelenkt.

DISPLAY IS und PRINTER IS Deklarationen bleiben gültig, bis:

- Sie einen neuen ASSIGN IO Befehl ausführen.
- Sie einen neuen DISPLAY IS oder PRINTER IS Befehl ausführen.
- Sie den Befehl OFF IO (Seite 130) eingeben.
- Der Informationsfluß in der Schleife unterbrochen wird (Seite 130).

Anzeigen und Drucken von Information (DISP, PRINT)

Mit zwei BASIC-Anweisungen können Sie Information auf HP-IL Einheiten anzeigen und ausdrucken.

```
DISP [Anzeigeliste] [;]
```

```
PRINT [Druckliste] [;]
```

DISP Anweisungen werden auf die Anzeige des HP-75 und auf die derzeitigen DISPLAY IS Einheiten gelenkt, während PRINT Anweisungen auf die derzeitigen PRINTER IS Einheiten gelenkt werden.

Beispiel:

```
>print 5555
```

Alle PRINTER IS Einheiten drucken 5555.

Wenn die Schleife keinen zugewiesenen Drucker enthält, führen die Anzeige des HP-75 und alle anderen Anzeigeeinheiten alle Druckeroperationen aus.

Manchmal ist es wünschenswert, Informationen mehrerer DISP oder PRINT Anweisungen in einer einzigen Zeile auszugeben. In diesem Fall muß die erste DISP oder PRINT Anweisung mit einem Semikolon abgeschlossen werden.

Beispiel:

```
>disp 'yes';
```

Das Semikolon unterdrückt den Wagenrücklauf/Zeilenvorschub, der normalerweise einer DISP Anweisung folgt.

Nachfolgende Anzeige- oder Druckinformationen werden solange in die gleiche Zeile geschrieben, bis ein Wagenrücklauf/Zeilenvorschub auftritt. Die Befehle DELAY und WIDTH regulieren die Verzögerungsrate und Zeilenlänge der

angezeigten Information; der Befehl PWIDTH bestimmt die Zeilenlänge der Druckerinformation (Seite 39). Programmieranwendungen der Anweisungen DISP und PRINT finden Sie im Abschnitt 11, auf Seite 166.

Führen Sie ASSIGN IO * oder ASSIGN IO '' aus, wenn Sie alle vorhergegangenen Schleifeninformationen im HP-75 löschen wollen. Danach werden alle Ausgaben auf die Anzeige gelenkt, und der von HP-IL Einheiten beanspruchte Systemspeicher steht wieder zur Verfügung.

Ein- und Ausschalten der Schleife (OFF IO, RESTORE IO)

Der Befehl OFF IO unterbindet die Kommunikation zwischen dem HP-75 und der Peripherie zeitweilig. Jeder nach OFF IO eingegebene HP-IL Befehl erzeugt Fehler 60 -RESTORE IO needed – und der HP-IL Befehl wird nicht ausgeführt. RESTORE IO ist der einzige HP-IL Befehl, der nach OFF IO ausgeführt werden kann.

```
OFF IO
```

Beispiel:

```
>offio
```

Alle Drucker- und Anzeigeinformationen erscheinen nur noch in der Anzeige des HP-75.

Benutzen Sie den Befehl OFF IO, bevor Sie Peripheriegeräte ausschalten oder aus der Schleife entfernen, um die derzeitigen Zuweisungen zu erhalten.

Der Befehl RESTORE IO stellt die Kommunikation zwischen dem HP-75 und der Peripherie nach einem OFF IO Befehl wieder her.

```
RESTORE IO
```

Schalten Sie alle Geräte ein und verbinden Sie sie zu ihrer ursprünglichen Konfiguration in der Schleife, bevor Sie RESTORE IO ausführen.

Beispiel:

```
>restore io
```

Die Peripherieeinheiten werden entsprechend ihren ursprünglichen Zuweisungen wieder aktiviert.

Der Befehl RESTORE IO dient dazu, ein HP-IL System in der vorhergehenden Konfiguration wieder zu aktivieren, und kann nicht einwandfrei funktionieren, wenn diese Konfiguration verändert wurde. Wenn die gleiche Anzahl von Geräten, aber in verschiedener Reihenfolge, angeschlossen ist, oder wenn mehr als die ursprüngliche Anzahl von Geräten angeschlossen sind, erkennt der HP-75 die Veränderung nicht, und die Einheiten können unerwartete Reaktionen auf HP-IL Befehle zeigen. Wenn weniger Einheiten angeschlossen sind, als sich Einheitscodes im Speicher befinden, erzeugt RESTORE IO Fehler 59 – too many names – und es erfolgt keine Wiederzuweisung der Einheiten. Wenn ein Fehler auftritt, müssen Sie entweder die ursprünglichen Geräte in ihrer ursprünglichen Reihenfolge wieder anschließen, oder sie müssen das derzeitige HP-IL System mit einem ASSIGN IO Befehl neu initialisieren.

Stellen Sie sicher, daß alle HP-IL Einheiten eingeschaltet und richtig angeschlossen sind, wann immer der HP-75 auf sie zugreift, – zum Beispiel bei der Benutzung einer PRINTER IS Einheit, beim Zugriff auf einen File im Massenspeicher, bei der Ausführung von OFF IO, RESTORE IO und BYE. Wenn diese Geräte nicht verfügbar sind, wartet der HP-75 auf eine Reaktion. Lesen Sie dazu den folgenden Abschnitt.

Unterbrechungen der Übertragung (ATTN, RESTORE IO, STANDBY OFF, STANDBY ON)

Die Zuweisung der Einheitscodes bleibt beim Ausschalten des HP-75 erhalten. Beim Wiedereinschalten nehmen die Einheiten ihre vorhergehende Funktion wieder an. Der HP-75 hat jedoch keine Kontrollmöglichkeit über den Betriebszustand der Einheiten in der Schleife.

Die Verbindung des Computers mit allen HP-IL Einheiten wird unterbrochen, wenn ein HP-IL Kabel entfernt wird, wenn ein Gerät stromlos wird, oder wenn eine Einheit ausgeschaltet wird, bevor ein OFF IO Befehl ausgeführt wurde. Danach reagiert die Anzeige möglicherweise nicht mehr auf das Drücken von Tasten, und Fehler 58 – loop timeout – kann angezeigt werden. Wenn Sie nach einem Ausschalten des HP-75 die Schleife abbauen wollen, sollten Sie daran denken, vor dem Ausschalten des Computers OFF IO auszuführen.

Drücken Sie nach einem Übertragungsfehler ATTN, um das HP-IL System zu unterbrechen; es wird Fehler 58 – loop timeout – angezeigt, wenn nicht bereits geschehen. Prüfen Sie dann die Kabel, Schalter und Stromversorgungen und beheben Sie die Ursache des Problems. Führen Sie dann Ihren HP-IL Befehl nochmals aus; oder führen Sie RESTORE IO aus, wenn ein ausgeschaltetes HP-IL Gerät die Ursache der Unterbrechung war. Wenn der HP-75 auf das Tastenfeld nicht reagiert, unterbrechen Sie und verbinden Sie danach wieder eine HP-IL Kabelverbindung und geben Sie den Befehl RESTORE IO ein.

Während einer HP-IL Operation, an der eine externe Drucker- oder Massenspeichereinheit beteiligt ist, bleibt der die Operation erzeugende Befehl in der Anzeige, solange das Gerät «beschäftigt» ist. Nachdem Sie zum Beispiel plist RTN eingegeben haben, bleibt der Befehl PLIST solange in der Anzeige, bis der externe Drucker für neue Anweisungen verfügbar ist.

Wenn die Schleife in der Stellung STANDBY OFF nach dem Befehl ASSIGN IO oder RESTORE IO unterbrochen wird, zeigt der HP-75 loop timeout an, nachdem er bis zu 30mal die Einheitszuweisung durchzuführen versucht und bei jedem erfolglosen Versuch ein Tonsignal erzeugt hat. Wenn Sie die Schleife sofort wieder schließen, wird der Befehl normal ausgeführt.

Wenn eine richtig angeschlossene DISPLAY IS oder PRINTER IS Einheit mehr als 10 Sekunden für die Anzeige oder das Ausdrucken eines einzelnen Zeichens benötigt (was normalerweise nicht der Fall sein wird), führt der HP-75 nach dieser Zeitspanne einen OFF IO Befehl aus und setzt den Betrieb ohne das HP-IL System fort. Prüfen Sie in einem solchen Fall die Einheiten in der Schleife und führen Sie RESTORE IO oder einen anderen HP-IL Befehl durch, um die Einheitszuweisungen wiederherzustellen. Wenn Sie wollen, daß der HP-75 länger als 10 Sekunden auf eine Antwort der Einheit wartet, müssen Sie Standby on RTN eingeben. Ist STANDBY ON gesetzt, wartet der Computer unbeschränkt lange darauf, daß die Drucker- oder Anzeigeeinheit ein Zeichen ausgibt, oder daß Sie die Taste ATTN drücken. Bei STANDBY ON wartet der Computer auch unbeschränkt lange auf die Antwort der Einheiten nach den Befehlen ASSIGN IO und RESTORE IO.

Wenn während einer HP-IL Operation ein Fehler auftritt, können Sie im allgemeinen davon ausgehen, daß die Operation nicht vollständig ausgeführt wird. Wenn die HP-IL zum Beispiel beim Kopieren eines Files auf Massenspeicher unterbrochen wird, wird ein Fehler angezeigt; der File ist lückenhaft und sollte gelöscht werden.

Zurücksetzen von Peripheriegeräten (CLEAR LOOP)

Der Befehl CLEAR LOOP setzt alle Einheiten in der Hewlett-Packard Interface Loop zurück. Auf diese Weise werden alle Einheiten in einen Bereitschaftszustand gebracht.

```
CLEAR LOOP
```

Jede Einheit reagiert ihrer Konstruktion entsprechend auf den Befehl CLEAR LOOP. Das Video-Interface HP82163 zum Beispiel löscht die Anzeige und setzt den Cursor in die «Homeposition» in die linke obere Ecke zurück; der Thermodrucker HP82162A setzt den Druckerkopf auf Spalte 1, löscht den Druckerbuffer und schaltet auf einfache Zeilenschaltung und Linksanschlag; das Digitalkassettenlaufwerk HP82161A spult die Kassette zurück. Die Reaktion einer bestimmten Einheit finden Sie im Benutzerhandbuch der Peripherieeinheit unter «Zurücksetzen der Einheit».

Sie können die Einheiten einzeln zurücksetzen, wenn Sie deren Einheitscodes im Befehl CLEAR spezifizieren.

```
CLEAR [:Einheitscode [;Einheitscode...]]
```

Beispiel:

```
>clear ':p'
```

Mit RTN wird nur Einheit P zurückgesetzt.

Massenspeicheroperationen

Massenspeicheroperationen dienen zur Speicherung und zum Zurücklesen von Informationen. Drei Komponenten sind an Massenspeicheroperationen beteiligt:

- Eine Massenspeicher*einheit*, das kann ein Magnetkartenleser, ein Kassettenlaufwerk oder ein Diskettenlaufwerk sein.
- Ein Massenspeicher*medium*, das eine Magnetkarte, eine Bandkassette oder eine Diskette sein kann.
- Ein Interface, das die Kommunikation des Computers mit der Massenspeichereinheit ermöglicht.

Auf den folgenden Seiten werden Massenspeicheroperationen anhand des Digital-Kassettenlaufwerks HP82161A als Massenspeicher*einheit*, und einer Bandkassette als Massenspeicher*medium* erläutert; als Interface dient die HP-IL.

Jeden File im Speicher, der auf Magnetkarte kopiert werden kann, können Sie auch auf ein Massenspeichermedium kopieren. Sie können Files im Massenspeicher ebenfalls katalogisieren, duplizieren, mit Paßwort sichern, umbenennen und löschen.

Die Massenspeichereinheit (ASSIGN IO)

Nachdem Sie einer Massenspeichereinheit mit dem Befehl ASSIGN IO einen Einheitscode zugewiesen haben, ist das Gerät einsatzbereit. In den folgenden Beispielen wird mit dem Einheitscode CA das Kassettenlaufwerk spezifiziert; Sie können in einer HP-IL Schleife aber durchaus mehrere Massenspeichereinheiten betreiben.

Beachten Sie, daß Sie Fehler 92 – dev not mass mem – erhalten, wenn Sie ein nicht dazu geeignetes Gerät für eine Massenspeicheroperation aufrufen; die Operation wird nicht ausgeführt. Wenn Sie eine Massenspeichereinheit als DISPLAY IS oder PRINTER IS deklarieren, wird die Anzeige- oder Druckerinformation an die Massenspeichereinheit geleitet (die Ergebnisse sind ohne Bedeutung); darauf folgende Massenspeicherbefehle erzeugen die Fehlermeldung 63 – invalid filespec – in der Anzeige.

> **Wichtig:** Wenn eine der Operationen INITIALIZE, COPY von Speicher auf Massenspeicher, RENAME oder PURGE unterbrochen wird, muß das Medium möglicherweise neu initialisiert werden. Stellen Sie sicher, daß alle Einheiten der Schleife eine verläßliche Stromversorgung besitzen, und drücken Sie nicht ATTN, und unterbrechen Sie keine HP-IL Kabelverbindung, solange eine Massenspeichereinheit eine Schreiboperation ausführt.

Vorbereitung des Massenspeichermediums (INITIALIZE)

Jedes Massenspeichermedium muß mindestens einmal initialisiert werden, um ein Fileverzeichnis und ein Format für die zu speichernde Information einzurichten. Bereiten Sie ein neues Medium mit dem Befehl INITIALIZE auf dessen Gebrauch vor.

> **Hinweis:** Wenn das Medium bereits zuvor benutzt worden ist, sollten Sie vor der Initialisierung dessen Inhalt überprüfen, wie auf Seite 134 beschrieben. Die Initialisierung eines Speichermediums löscht alle zuvor dort gespeicherten Informationen vollständig.

```
INITIALIZE ':Einheitscode' [,Anzahl der Fileeinträge]
```

Das Initialisieren einer Bandkassette im Digital-Kassettenlaufwerk HP82161A nimmt etwa 5 Minuten in Anspruch.

Jeder File, den Sie aufzeichnen, erzeugt einen Eintrag in den Katalog des Mediums. Wenn Sie bei INITIALIZE nicht die Anzahl der Fileeinträge vorgeben, kann der Katalog maximal 128 Fileeinträge aufnehmen.

Sie können einen beliebigen numerischen Ausdruck spezifizieren, der auf einen ganzzahligen Wert größer Null gerundet wird. Mit Werten kleiner als 1 erhalten Sie Fehler 11 – `arg out of range`.

Beim Digital-Kassettenlaufwerk HP82161A sind maximal 453 Katalogeinträge möglich. Werte größer als 453 erzeugen Fehler 11 – `argout of range`. Der Katalogplatz wird in Blocks von acht Fileeinträgen zur Verfügung gestellt; der Katalog enthält folglich immer ein Vielfaches von acht Fileeinträgen, – mindestens so viele, wie Sie spezifiziert haben.

Ein nicht initialisiertes Massenspeichermedium erzeugt bei einem Zugriffsversuch Fehler 96 – `invalid medium`. Dies kann auch erfolgen, wenn das Medium von einem anderen Computer als dem HP-75 initialisiert wurde, wenn das Medium abgenutzt oder beschädigt ist, oder wenn das Kassettenlaufwerk während der Initialisierung unterbrochen wird.

Massenspeicherfiles

Im Massenspeicher wird Information in Form von Files (Ansammlungen von Datensätzen) abgelegt, auf die über Filenamen zugegriffen wird. Folgende Arten von Files können auf einem Massenspeichermedium abgelegt werden:

- BASIC-Files, mit einem `B` gekennzeichnet. (Private BASIC-Files, `PB`, sind auf Magnetkarten, nicht aber auf einem Massenspeichermedium, möglich.)
- Textfiles, mit einem `T` gekennzeichnet.
- Terminfiles, mit einem `A` gekennzeichnet.
- Tastenfiles, ebenfalls mit einem `T` gekennzeichnet.
- LEX-Files, mit einem `L` gekennzeichnet.
- Austauschfiles, mit einem `I` gekennzeichnet.
- Unbekannte Files, gekennzeichnet durch `?`. Files des Typs `?` können katalogisiert, auf dem Medium dupliziert, umbenannt und gelöscht, aber nicht in den Speicher eingelesen werden.

Sie können individuelle Files auf dem Massenspeichermedium über einen der beiden folgenden Filespezifikatoren aufrufen:

Filename : *Einheitscode*
Filename : *Einheitscode*/*Paßwort*

Der *Filename* muß ein zulässiger Filename (siehe Abschnitt 3) sein. Der *Einheitscode* ist der der Massenspeichereinheit im letzten `ASSIGN IO` zugewiesene Name. Das *Paßwort* beschränkt den Zugriff auf den File im Massenspeicher auf Besitzer dieses Codes. Das Paßwort kann aus bis zu vier Buchstaben und Ziffern bestehen.*

Der gesamte Filespezifikator steht in Anführungszeichen. Beispiele für zulässige Massenspeicher-Filespezifikatoren sind `'data:ca'`, `'numcalc:ca'` und `'seth:ca/123'`. (Die reservierten Worte `keys`, `appt` und `workfile` sind als Filenamen in Filespezifikatoren erlaubt; sie werden jedoch bei Massenspeicherperationen als gewöhnliche Filenamen behandelt.)

Hinweis für den fortgeschrittenen Benutzer: Sie können für Filespezifikatoren anstelle von Strings in Anführungszeichen auch Stringvariablen und -ausdrücke verwenden. Filespezifikatoren werden mit dem zweiten Anführungszeichen oder mit dem ersten Leerzeichen nach dem Einheitscode bzw. Paßwort abgeschlossen.

*Sie können allen vom HP-75 erzeugten Files außer Austauschfiles ein Paßwort zuweisen. Wenn Sie für einen Austauschfile in einem beliebigen Massenspeicherbefehl ein Paßwort spezifizieren, erhalten Sie Warnung 66 – `invalid password` – aber der Befehl kann ausgeführt werden; das Paßwort wird ignoriert.

Beachten Sie, daß ein Paßwort den File vor dem Kopieren, nicht aber vor dem Löschen oder vor einem Umbenennen, schützt. Nachdem das Paßwort richtig eingegeben und der File in den Speicher eingelesen wurde, kann der neue File im Speicher editiert werden.

Andere Computer erlauben Filenamen mit mehr als acht Zeichen. Ein Katalogeintrag auf einem Speichermedium kann bis zu 10 Zeichen pro Filenamen enthalten; der HP-75 arbeitet in Massenspeicherbefehlen jedoch nur mit acht Zeichen und meldet einen Fehler, wenn mehr als acht Zeichen für einen Filenamen eingegeben werden.

Wenn sich ein Filespezifikator auf einen auf dem Massenspeichermedium nicht existierenden File bezieht, erhalten Sie die Fehlermeldung 62 – `file not found`.

Inhaltsverzeichnis des Mediums (CAT)

Wenn Sie die Katalogeinträge des sich derzeit in der Massenspeichereinheit befindenden Mediums anzeigen wollen, geben Sie dazu den Namen der Einheit im Befehl `CAT` ein.

```
CAT  ' :Einheitscode'
```

Beispiel:

```
>cat ':ca'
```

Anzeige der Katalogeinträge des Mediums in der Massenspeichereinheit `CA`.

Wenn Sie [RTN] drücken, wird die Kopfzeile des Katalogs angezeigt:

```
Name    Type Len Time Date
```

Bleibt für die momentane Verzögerungszeit in der Anzeige, dann...

```
ANFANG B 4096 12:30 02/09/83
```

Der Katalogeintrag des ersten Files auf dem Medium wird angezeigt.

Mit den Tasten [↑] und [↓] können Sie die Katalogeinträge nacheinander anzeigen. Nach jedem Tastendruck muß das Massenspeichermedium wieder neu positioniert und gelesen werden, was eine geringe Verzögerung bedingt. Mit [SHIFT] [↑] erhalten Sie den ersten und mit [SHIFT] [↓] den letzten Katalogeintrag des Inhaltsverzeichnisses. Auch die Tasten [←] und [→] und die Umschalttasten [SHIFT] und [CTL] sind aktiv, während Sie Katalogeinträge in der Anzeige haben. Alle anderen Tasten brechen die Operation ab und erzeugen den Cursor und die Eingabeaufforderung in der Anzeige.

Katalogeinträge werden in der gleichen Reihenfolge angezeigt, wie sie auf dem Medium aufgezeichnet wurden. Bevor der HP-75 einen File abspeichert, sucht er den Katalog nach der ersten freien Stelle auf dem Medium ab, die den gesamten File aufnehmen kann. Ein leeres, aber initialisiertes Medium zeigt bei der Ausführung von `CAT` nur die Kopfzeile des Katalogs an.

Sie können im Befehl `CAT` auch einen einzelnen Massenspeicherfile spezifizieren.

```
CAT 'Filespezifikator'
```

Diese Form des Befehls zeigt nur den Katalogeintrag des spezifizierten Files an; ein Paßwort ist nicht erforderlich. Wenn der HP-75 `file not found` anzeigt, existiert der spezifizierte Name nicht im Katalog des Mediums.

Kopieren von Files auf und von Massenspeicher (COPY)

Der Befehl COPY ermöglicht Ihnen, einzelne Files vom Speicher auf ein Massenspeichermedium oder von einem Massenspeichermedium in den Speicher zu kopieren. Kopieren Sie nicht geschützte Files mit einem der folgenden Befehle vom Speicher auf ein Massenspeichermedium:

```
COPY TO ':Einheitscode'
COPY 'Filename' TO ':Einheitscode'
COPY TO 'Filespezifikator'
COPY 'Filename' TO 'Filespezifikator'
```

Beispiele:

```
>copy to ':ca'
```

Kopiert den momentanen File auf die Massenspeichereinheit. Zeit und Datum werden übernommen.

```
>copy "file2" to ':ca'
```

Kopiert File2 im Speicher auf die Massenspeichereinheit. Name, Zeit und Datum werden übernommen.

```
>copy to 'neufile:ca'
```

Kopiert den momentanen File als NEUFILE auf den Massenspeicher.

```
>copy 'file3' to 'file4:ca'
```

Kopiert FILE3 als FILE4 auf den Massenspeicher. Neuer Name, neue Zeit, neues Datum.

Zeit und Datum von Quell- und Zielfile stimmen überein, wenn die Namen der Files ebenfalls übereinstimmen. Bei verschiedenen Filenamen wird der neue File nach den Angaben der Systemuhr bei der Ausführung des COPY Befehls neu datiert.

Das Kopieren eines Files auf und von Massenspeicher kann je nach Länge des Files eine oder zwei Minuten in Anspruch nehmen.

Wenn Sie in einem Befehl, der einen File auf Massenspeicher kopiert, ein Paßwort spezifizieren, wird der File mit dem Paßwort gesichert abgespeichert.

Beispiele:

```
>copy 'file5' to 'file6:ca/12'
```

Kopiert FILE5 im Speicher auf einen Massenspeicherfile FILE6 und sichert diesen mit dem Paßwort 12

```
>copy appt to 'januar:ca/12'
```

Kopiert den appt-File auf den Massenspeicherfile JANUAR und sichert ihn mit dem Paßwort 12.

Private BASIC-Files (Typ PB) im Speicher können nicht auf Massenspeicher kopiert werden.

Hinweis: Wenn ein Kopierbefehl einen File im Massenspeicher ablegen soll, dessen Name dort schon existiert, wird der File auf dem Medium mit der Information des Files im Speicher überschrieben. Danach kann der File an einer anderen Stelle im Katalog verzeichnet sein.

Zum Kopieren eines Files vom Massenspeicher in den Speicher können Sie die folgende Form des COPY Befehls verwenden:

```
COPY 'Filespezifikator' TO 'Filename'
COPY ':Einheitscode' TO 'Filename'
```

Beispiele:

```
>copy 'file4:ca' to 'file8'
```

Kopiert FILE4 vom Massenspeicher auf FILE8 im Speicher. Neue Zeit, neues Datum.

```
>copy 'file6:ca/12' to 'file7'
```

Kopiert FILE6, der mit dem Paßwort 12 gesichert ist, vom Massenspeicher auf den Speicherfile FILE7. Neue Zeit, neues Datum.

Wenn ein File im Massenspeicher mit einem Paßwort gesichert ist, muß dieses im Filespezifikator enthalten sein; andernfalls erhalten Sie die Fehlermeldung 66 – invalid password – und der Kopiervorgang wird nicht gestattet. (Der HP-75 ignoriert Paßwörter, die für nicht gesicherte Files eingegeben werden.)

Wenn für einen einzulesenden File nicht genügend Platz im Speicher zur Verfügung steht, meldet der HP-75 Fehler 16 – not enough memory –, und der Kopiervorgang wird nicht ausgeführt.

Wenn ein Terminfile von einem Massenspeichermedium in den appt- File im Speicher übertragen wird, werden die Termine des Massenspeicherfiles chronologisch im appt-File *eingeordnet*.

Beispiel:

```
>copy 'januar:ca/12' to appt
```

Ordnet die Termine des Files im Massenspeicher in den appt-File ein.

Alle anderen Files, die in den Speicher kopiert werden sollen, müssen eigene Filenamen besitzen; andernfalls erhalten Sie Fehler 64 – duplicate name – und der Kopiervorgang wird nicht ausgeführt. Ein File im Massenspeicher, der in den appt-File kopiert werden soll, muß ein Terminfile (A) sein; ein File im Massenspeicher, der auf den File keys kopiert werden soll, muß ein Textfile (T) sein; andernfalls erhalten Sie die Fehlermeldung 68 – wrong file type. Die Tastendefinitionen werden wirksam, sobald die copy to keys Operation abgeschlossen ist.

Wenn ein Kopiervorgang in den Massenspeicher mit der Taste [ATTN] unterbrochen wird, wird der teilweise kopierte File vom Massenspeicher, wenn möglich, wieder gelöscht. Der Cursor und die Eingabeaufforderung erscheinen nach kurzer Verzögerung wieder. Drücken Sie *nicht* noch einmal [ATTN], bevor der Löschvorgang abgeschlossen ist, andernfalls muß das Massenspeichermedium möglicherweise neu initialisiert werden. Wenn ein Kopiervorgang in den Speicher unterbrochen wird, wird das Filefragment im Speicher sofort gelöscht.

Duplizieren von Files im Massenspeicher (COPY)

Mit dem Befehl COPY können Sie einen File im Massenspeicher duplizieren.

```
COPY 'Filespezifikator' TO 'Filespezifikator'
```

Beispiele:

```
>copy 'file8:ca' to 'file9:ca'
```

Kopiert FILE8 in FILE9 auf dem selben Speichermedium.

```
>copy 'file8:c1' to ':c2/1'
```

Kopiert FILE8 auf einer Massenspeichereinheit in FILE8 einer anderen Massenspeichereinheit. Der neue File ist mit dem Paßwort 1 gesichert.

Wenn der duplizierte File auf der Massenspeichereinheit erzeugt wird, auf der sich auch das Original befindet, muß der Zielname vom Quellnamen verschieden sein. Wenn Sie den File auf eine andere Massenspeichereinheit duplizieren, wird der gleiche Filename benutzt, falls Sie nicht einen anderen spezifizieren. Wenn der Filename auf der zweiten Einheit schon existiert, erhalten Sie Fehler 64 – duplicate name – und die Kopieroperation wird nicht ausgeführt.

Wie bei anderen Kopieroperationen sind bei gleichen Filenamen Zeit und Datum identisch. Zumindest ein Filename muß spezifiziert werden.

Umbenennen von Files im Massenspeicher

Sie haben drei Formen des Befehls RENAME zur Umbenennung von Files auf einem Massenspeichermedium zur Verfügung.

```
RENAME  'Filespezifikator'  TO  'Filename'
RENAME  'Filename'  TO  'Filespezifikator'
RENAME  'Filespezifikator'  TO  'Filespezifikator'
```

Alle drei Formen des Befehls sind in ihrer Wirkung identisch.

Beispiel:

```
>rename 'file0:ca' to 'file10'
```

Ändert den Namen von FILE0 auf Massenspeicher in FILE10.

Der Befehl RENAME verlangt kein Paßwort und ignoriert gegebenenfalls angegebene Paßworte.

> **Wichtig:** Wenn eine RENAME Operation unterbrochen wird, muß das Medium möglicherweise neu initialisiert werden.

Löschen von Files im Massenspeicher

Mit dem Befehl PURGE können Sie einen File vom Massenspeichermedium löschen.

```
PURGE  'Filespezifikator'
```

Beispiel:

```
>purge 'file10:ca'
```

Entfernt FILE10 aus dem Katalog des Speichermediums.

Der Befehl PURGE benötigt kein Paßwort; wenn Sie dennoch eines angeben, wird es ignoriert.

> **Wichtig:** Wenn eine PURGE Operation unterbrochen wird, muß das Medium möglicherweise neu initialisiert werden.

Packen des Massenspeichermediums (PACK)

Beim Löschen von Files auf einem Medium hinterlassen diese «Lücken» zwischen den Nachbarfiles. Bei der Aufzeichnung anderer Files auf das Medium füllt der HP-75 diese Lücken mit Files gleicher oder kleinerer Länge. Irgendwann steht einmal nicht mehr genügend Platz für neue Files zur Verfügung, und Sie erhalten den Fehler 95 – medium full – beim Beginn eines Kopiervorgangs. Sie können dann mit den Befehl PACK wieder Speicherplatz für neue Files zur Verfügung stellen.

```
PACK  ':Einheitscode'
```

Der *Einheitscode* gibt an, auf welcher Massenspeichereinheit sich das zu packende Medium befindet.

Beispiel:

```
>pack ':ca'
```

Die auf der Einheit CA existierenden Files werden neu auf dem Medium angeordnet, um optimale Ausnutzung des Speicherplatzes zu gewährleisten.

Der Packvorgang kann einige Minuten bis zu einer Stunde dauern, je nach Anzahl und Länge der Files auf dem Medium.

> **Wichtig:** Wenn eine PACK-Operation unterbrochen oder Fehler 97 – invalid pack – gemeldet wird, muß das Medium mit großer Wahrscheinlichkeit neu initialisiert werden. Versuchen Sie nicht, COPY oder CAT auszuführen. Die Daten auf dem Medium sind unbrauchbar, und der HP-75 muß unter Umständen vollständig zurückgesetzt werden, wenn solche Daten in den Speicher eingelesen werden. Stellen Sie sicher, daß die Schleifeneinheiten verläßlich mit Strom versorgt sind, bevor Sie PACK durchführen. Drücken Sie nicht [ATTN], während eine Packoperation ausgeführt wird.

Beachten Sie, daß das Packen eines Mediums das Medium in der Massenspeichereinheit nicht unbeträchtlich beansprucht. Packen Sie das Medium nur, wenn es unbedingt erforderlich ist, um die Lebensdauer des Mediums nicht unnötig zu vermindern.

Information für den fortgeschrittenen Benutzer

Sonderzeichen

In Abschnitt 2 wurde der vollständige Zeichensatz des HP-75, bestehend aus 256 Buchstaben, Ziffern, Punktuationszeichen, Symbolen und anderen Sonderzeichen, eingeführt. Dem Zeichensatz des HP-75 sind Dezimalcodes von 0 bis 255 zugeordnet.

Alle HP-IL Anzeige- und Druckereinheiten zeigen bzw. drucken die Zeichen 32 bis 125; die meisten Anzeige- und Druckereinheiten erkennen und beantworten jedoch auch Zeichen außerhalb dieses Bereichs. Das Videointerface HP 82163 zum Beispiel interpretiert die Codes 160 bis 254 als Inversvideozeichen (schwarz auf weiß). [CTL] [J], (Dezimalcode 10), erzeugt in der Anzeige des HP-75 und auf dem Videointerface einen Zeilenvorschub. Sie finden die Reaktionen einzelner Peripheriegeräte auf Zeichen mit den Dezimalcodes 0 bis 31 und 127 bis 255 im jeweiligen Benutzerhandbuch.

Escapecodes ([CTL] [BACK], CHR$(27), [CTL] [↑], [CTL] [↓])

Das Escapezeichen ESC (Dezimalcode 27) aktiviert spezielle HP-IL Operationen. Dieses Zeichen wird nicht angezeigt.

Sie können ESC mit CHR$(27) in einem DISP- oder PRINT-Befehl spezifizieren; damit wird bei der Ausführung der Anweisung ESC auf das entsprechende Ausgabegerät gerichtet, und gleichzeitig kann eine solche Programmzeile normal gelistet und angezeigt werden.

Wenn Sie ESC mit [CTL] [BACK] auf dem Tastenfeld erzeugen, zeigt die Anzeige >, und der Cursor verschwindet – das Drücken der nächsten Taste bringt den Cursor wieder zur Anzeige. Aber eine Programmzeile, die ein ESC-Zeichen (als [CTL] [BACK]) enthält, kann nicht normal gelistet oder angezeigt werden, weil ESC und das oder die nachfolgenden Zeichen als *Escapecode* interpretiert werden.

Ein *Escapecode* ist ein aus dem Escapezeichen und einem oder mehreren darauffolgenden Zeichen bestehender Zeichenstring. Wenn ein Escapecode an eine DISPLAY IS oder eine PRINTER IS Einheit gerichtet wird, ignoriert die Einheit den gesamten Escapecode oder einen Teil davon oder interpretiert ihn als Anzeige- oder Druckeranweisung. Ein Escapecode kann in einem längeren Stringausdruck enthalten sein.

Wenn Sie zum Beispiel ' [CTL] [BACK] E' [RTN] drücken, löscht das Videointerface den Bildschirm und positioniert den Cursor in die linke obere Ecke. Die Eingabe von disp chr$(27) &'E'; [RTN] hat die gleiche Wirkung.

Zwei weitere Tastenfolgen senden Escapecodes an DISPLAY IS Einheiten:

[CTL] [↑] sendet ESC S; die DISPLAY IS Einheiten «rollen» die Anzeige um eine Zeile nach oben. [CTL] [↓] erzeugt ESC T; die DISPLAY IS Einheiten «rollen» um eine Zeile nach unten. Die Anzeige des HP-75 wird weder von [CTL] [↑] noch von [CTL] [↓] beeinflußt.

Im Benutzerhandbuch zu Ihrem Peripheriegerät finden Sie weitere Informationen über die jeweiligen Escapecodes der Einheit.

Escapecodes zur Anzeigesteuerung

Die Anzeige des HP-75 reagiert als DISPLAY IS Einheit auf 12 Escapecodes. Die 32 Fensterpositionen werden über die Nummern 0 bis 31 identifiziert.

Escapecode	Antwort des Anzeigefensters
ESC C	Bewegt den Cursor um eine Stelle nach rechts.
ESC D	Bewegt den Cursor um eine Stelle nach links.
ESC E	Setzt den Cursor auf Spalte 0 und löscht die Anzeige.
ESC G	Setzt den Cursor auf Spalte 0.
ESC J und ESC K	Löscht die Anzeige ab der momentanen Cursor-Position.
ESC O und ESC P	Löscht das Zeichen an der momentanen Cursorposition und schiebt alle nachfolgenden Zeichen nach links.
ESC <	Schaltet den Cursor aus.
ESC >	Schaltet den Cursor ein.
ESC %*cr*	Setzt den Cursor auf die durch den Dezimalcode von *c* spezifizierte Spalte. (*r* wird vom HP-75 ignoriert).

Acht dieser Escapecodes haben ähnliche Wirkung auf das Videointerface HP82163 (siehe Anhang D, Seite 294).

Beispiel:

```
10 WIDTH INF

20 DISP CHR$(27);'%';CHR$(RND*32
);CHR$(7);


30 DISP CHR$(RND*93+33);


40 GOTO 20
```

Spezifiziert unbegrenzt viele Zeichen pro Zeile.

Setzt den Cursor auf eine zufällige Spaltenposition. (Die 7 spezifiziert Zeile 7 des Videointerface HP82163; Zeilenparameter haben auf das Anzeigefenster keine Auswirkung.)

Zeigt ein Zufallszeichen an. Das Semikolon unterdrückt den Wagenrücklauf/Zeilenvorschub.

Positioniert den Cursor neu und zeigt ein weiteres Zeichen an.

Bei Ausführung des Programms erhalten Sie diese oder ähnliche Anzeigen:

```
>D   U f kd 8E  4u2    !Q 30P= y
```

An den zufälligen Positionen entstehen solange zufällige Zeichen, bis Sie [ATTN] drücken.

Sie können die Anzeige und den Cursor mit der Tastenfolge [CTL] [BACK] direkt über das Tastenfeld kontrollieren. Mit der folgenden Tastenfolge wird der Cursor auf Spalte 29 der Anzeige gestellt:

' [CTL] [BACK] [SHIFT] [5] [CTL] [=] [Leertaste] ' ; [RTN]

Mit [SHIFT] [5] wird das Zeichen % angezeigt; der Zeichencode von [CTL] [=] ist 29, und die Leertaste setzt eine Null ein (andernfalls würde das nächste angezeigte Zeichen als Teil der Escapesequenz interpretiert werden).

Beachten Sie, daß mit dem Drücken von [CTL] [BACK] die Escapezeichen in den Eingabebuffer eingegeben werden, auch wenn das Zeichen selbst nicht angezeigt wird. Beachten Sie auch, daß die Sequenz in Form einer impliziten DISP Anweisung in Anführungszeichen eingeschlossen ist und mit einem Semikolon abgeschlossen wird, um die Steuerzeichen CR/LF (Wagenrücklauf/Zeilenvorschub), die normalerweise nach der Anzeige folgen würden, zu unterdrücken. (CR/LF würde den Cursor unmittelbar wieder an den Zeilenanfang zurücksetzen.) Über das Tastenfeld auszuführende Escapesequenzen müssen als DISP Anweisungen eingegeben werden. Steuern Sie den Cursor bei der Ausführung von Programmen mit der Funktion CHR$. Das Ihrem HP-75 beiliegende Programm AUFGPST benutzt die Anzeige-escapecodes auf sehr effiziente Weise. Sie finden ein Listing des Programms AUFGPST in Anhang F.

End-of-Line Sequenzen (ENDLINE)

Nach der Ausführung einer PRINT Anweisung (falls diese nicht mit einem Semikolon endet) sendet der HP-75 zwei weitere Zeichen – einen Wagenrücklauf und einen Zeilenvorschub (CR/LF = *Carriage Return/Line Feed*) – an die PRINTER IS Einheiten. Auch bei einer PLIST Operation wird am Ende jeder Zeile, oder sobald die Zeilenlänge die momentane PWIDTH Einstellung überschreitet, ein CR/LF erzeugt.

Sie können mit dem Befehl END LINE die normale CR/LF End-of-Line Zeichenfolge ändern.

ENDLINE [*Stringausdruck mit 0 bis 3 Zeichen*]

In einem ENDLINE Befehl können bis zu drei Zeichen spezifiziert werden.

Beispiel:

```
>endline chr$(13)&chr$(10)&chr$(
10)
```

Erzeugt eine doppelte Zeilenschaltung nach jeder PRINT Anweisung.

Ein ENDLINE Befehl kann beliebige Zeichen des HP-75 enthalten; die Zeichen werden am Ende jeder Druckzeile ausgedruckt. Wenn Sie den Nullstring (`''` oder `""`) spezifizieren, wird die End-of-Line Sequenz unterdrückt, und PLIST oder PRINT Ausgaben werden als kontinuierlicher String (ohne Zeilenschaltung) ausgedruckt. Wenn Sie ENDLINE ohne Spezifikationen eingeben, wird die voreingestellte CR/LF-Sequenz wieder aktiviert.

ENDLINE Befehle haben keine Auswirkung auf Ausgaben mit LIST oder DISP. Die derzeitige ENDLINE Einstellung bleibt erhalten, bis ein weiterer ENDLINE Befehl ausgeführt wird. Mit `endline chr$(13)&chr$(10)` [RTN], oder einfach mit ENDLINE ohne Parameter erhalten Sie wieder die normale HP-75 End-of-Line Sequenz.

Programmieren von HP-IL Befehlen

Alle HP-IL Befehle sind programmierbar. Weiterhin können alle Parameter der HP-IL Befehle (wie Einheitscodes) über Stringvariablen oder Stringausdrücke spezifiziert werden.

Programme können Einheitscodes zuweisen, Anzeige- und Druckereinheiten deklarieren, die Ausgabe an Anzeige- und Druckereinheiten lenken, Files von und auf Massenspeicher kopieren, den HP-75 ein- und ausschalten und End-of-Line Sequenzen setzen.

Abschnitt 10

Belegen des Tastenfelds

Inhalt

Einführung

Eine der speziellen Eigenschaften des HP-75 ist das frei belegbare Tastenfeld. Sie können 194 Tasten und Tastenkombinationen neue Funktionen zuweisen. Folgende Tasten können Sie neu belegen:

- 60 nicht umgeschaltete Tasten (Beispiele: [Q], [←], [APPT])
- 59 umgeschaltete Tasten (Beispiele: [SHIFT] [5], [SHIFT] [←], [SHIFT] [ATTN])
- 61 Kontrolltasten (Beispiele: [CTL] [Z], [CTL] [8], [CTL] [LOCK])
- 14 *umgeschaltete* Kontrolltasten (Beispiel: [SHIFT] [CTL] [TAB])

Sie können *Tastenkombinationen* definieren, die die Tasten [SHIFT] oder [CTL] als Umschalttasten benutzen, obwohl [SHIFT] und [CTL] selber nicht umdefiniert werden können. Sie sollten Tastenfolgen,die das System zurücksetzen, wie zum Beispiel [SHIFT] [CTL] [CLR] (siehe Anhang C) nicht umdefinieren, da diese Ihren Computer bei jedem derartigen Versuch zurücksetzen.

Belegungen von Tasten und Tastenkombinationen sind für folgende Anwendungen nützlich:

- Eingabehilfen, um Eingaben über das Tastenfeld zu beschleunigen.
- Sofortausführungstasten für beliebige Operationen des HP-75 in allen drei Modi, EDIT, TIME und APPT.

Ein spezieller Textfile mit dem Namen keys legt die derzeitigen Tastenbelegungen im Speicher ab. Der keys File kann wie andere Textfiles editiert, umbenannt und von und auf Massenspeicher und Magnetkarten kopiert werden. Sie können eine Vielzahl selbst entworfener Tastenfelder im Speicher und auf Magnetkarten abspeichern und wieder benutzen.

Eingabehilfen (DEF KEY)

Eine Eingabehilfe ist eine Taste oder Tastenkombination, die einen Zeichenstring in der Anzeige erzeugt. Die Taste [FET] ist eine Eingabehilfe, da sie das Wort FETCH zur Anzeige bringt. Eingabehilfen können das Anzeigefenster löschen, Meldungen beliebiger Länge anzeigen und den Cursor irgendwo auf der Anzeigezeile positionieren.

Tastenbelegungen erfolgen mit Hilfe des Befehls DEF KEY.*

```
DEF KEY 'Tastenanzeigezeichen', 'Tastendefinition'[;]
```

Jede DEF KEY Deklaration definiert eine Taste oder Tastenkombination. DEF KEY ist programmierbar und kann im EDIT-Modus mit der Text- oder der BASIC-Eingabeaufforderung in der Anzeige ausgeführt werden.

Das *Tastenanzeigezeichen* ist das Zeichen, das zu der zu definierenden Taste oder Tastenkombination gehört. Die Anzahl von Zeichen in der *Tastendefinition* ist nur durch die Länge der Anzeige begrenzt. Beide Parameter stehen in Anführungszeichen (' ' oder " ") und sind durch ein Komma getrennt. Das optionale Semikolon (;) am Ende bestimmt, ob die Taste als Eingabehilfe oder als Sofortausführungstaste dient.

Im folgenden Beispiel wird eine triviale Eingabehilfe erzeugt: die Taste [Q] – nicht umgeschaltet- soll das Zeichen einer anderen Taste – der nicht umgeschalteten Taste [X] – erzeugen. Geben Sie die folgende Deklaration ein und schließen Sie mit [RTN] ab:

```
>def key 'q','x';
```

Ein Komma trennt die beiden Parameter. Diese DEF KEY Deklaration wurde mit einem Semikolon abgeschlossen.

```
>
```

Die Taste [Q] ist jetzt umdefiniert.

Wenn Sie nun die nicht umgeschaltete Taste [Q] drücken, wird ein x angezeigt. Beachten Sie, daß [SHIFT] [Q] noch immer ein Q anzeigt, und daß [CTL] [Q] das Zeichen Q erzeugt – diese Tastenkombinationen wurden durch die Tastenbelegung nicht geändert.

Sie können die ursprüngliche Tastenbelegung kurzzeitig wieder herstellen, wenn Sie [SHIFT] [I/R] (die Tastenfolge für Anzeigezeichen) vor der umdefinierten Taste drücken.

Beispiel:

1. Halten Sie [SHIFT] gedrückt, und drücken Sie dazu die Taste [I/R].
2. Geben Sie [SHIFT] [I/R] wieder frei.
3. Drücken Sie [Q]: ein kleines q wird angezeigt.

Wie Sie sehen, können Eingabehilfen sehr schnell erzeugt werden.

Beispiel: Ändern Sie die Taste [%] ([SHIFT] [5]) in eine Eingabehilfe für den Ausdruck LIST um. Löschen Sie mit [CLR] die Anzeige, und geben Sie ein:

```
>def key '%','LIST';
```

LIST wird wie eingegeben in der Anzeige erscheinen, da es in Anführungszeichen eingeschlossen wurde.

```
>
```

Die neue Tastenbelegung ist im Speicher abgelegt.

*Die Tasten [ATTN], [EDIT] und [RTN] und die Tastenkombination [SHIFT] [I/R] (Anzeigezeichen) können nur indirekt, nicht aber mit einer DEF KEY Deklaration, umdefiniert werden (siehe Seite 151).

Die Taste [%], oder [SHIFT] [5], zeigt nun LIST anstelle von % an. Wenn Sie die Taste gedrückt halten, erhalten Sie eine ganze Zeile voller LISTs in der Anzeige. Um stattdessen das Symbol % zu erhalten, müssen Sie [SHIFT] [I/R] und *dann* [SHIFT] [%] drücken.

Anzeigezeichen mit Dezimalcodes von 128 bis 255 erscheinen bei Drücken der Tasten, denen sie zugewiesen sind, nicht in der Anzeige, wenn die *Tastendefinition* nicht die Tastenkombination [SHIFT] [IR] enthält. Wenn Sie das Tastenzeichen von [SHIFT] [I/R] (◺) selbst erhalten wollen, müssen Sie [SHIFT] [I/R] [SHIFT] [I/R] eingeben.

Beispiel: Definieren Sie das große A mit Hilfe von DEF KEY in das unterstrichene große A um. Die *Tastendefinition* muß das Anzeigezeichen [SHIFT] [I/R] enthalten; Sie müssen also dreimal [SHIFT] [I/R] drücken – zweimal, um das Anzeigezeichen [SHIFT] [I/R] einzugeben, und einmal, um das A einzugeben. Die vollständige Tastenfolge für diese Definition lautet:

def key 'A', '[SHIFT] [I/R] [SHIFT] [I/R] [SHIFT] [I/R] [CTL] [TIME]';

Bevor Sie [RTN] drücken, sieht die Anzeige dann so aus:

```
>def key 'A',  '◺A';▮
```

```
>▮
```

Die neue Tastendefinition wird abgespeichert, wenn Sie [RTN] drücken.

Der keys-File (CAT KEYS)

Der File keys ist ein spezieller Textfile, der die Belegungen von Tasten des HP-75 speichert. Er ist ein Sonderfile, da er vom Betriebssystem des HP-75 erzeugt und verändert wird. Sie können den keys File jedoch genauso untersuchen und editieren wie jeden anderen File auch. Mit dem Befehl CAT KEYS erhalten Sie den Katalogeintrag des keys-Files:

```
>cat keys
```

```
keys     T  28 09:55 02/10/83
```

Der Katalogeintrag des keys Files mit Filenamen, Filetyp (T für Text), Filelänge (28 Bytes), Erzeugungszeit und -datum.

Die Zeit und das Datum im Katalogeintrag des keys-File zeigen, wann der File erzeugt wurde. Den keys File in diesem Abschnitt haben Sie in dem Moment erzeugt, als Sie bei der Definition der Taste [Q] die Taste [RTN] gedrückt haben.

Aufheben von Tastenbelegungen ([SHIFT] [I/R], DEF KEY, PURGE KEYS)

Sie können einer neu belegten Taste oder Tastenkombination auf vier verschiedene Weisen ihre ursprüngliche Funktion wieder zuweisen:

- Drücken Sie [SHIFT] [I/R], bevor Sie die Taste selbst drücken. Dies ist eine befristete Rückumwandlung.
- Deklarieren Sie die Originalfunktion mit dem Befehl DEF KEY.
- Ändern Sie den Namen des keys-Files (siehe Seite 147).
- Löschen Sie den gesamten keys-File; damit werden alle Tastenbelegungen gelöscht, der keys-File verschwindet aus dem Systemkatalog.

Beispiel: Weisen Sie der nicht umgeschalteten Taste [Q] mit Hilfe des Befehls DEF KEY wieder ihre Originalfunktion, Anzeige des kleinen q, zu. Die DEF KEY Deklaration muß zwei q's enthalten, deshalb müssen Sie zweimal [SHIFT] [I/R] drücken, für jedes q einmal:

```
>def key 'q', 'q':
```

```
>
```

Wenn Sie im Befehl DEF KEY einen Nullstring spezifizieren (eine Tastendefinition, die zwei Anführungszeichen und kein Zeichen dazwischen enthält), wird die betreffende Taste oder Tastenkombination desaktiviert.

Beispiel:

```
>def key "q","";
```

Wenn Sie [RTN] drücken, ist die Taste [Q] danach ohne jede Funktion. Denken Sie daran, der Taste wieder ihre ursprüngliche Funktion zuzuweisen.

Sofortausführungstasten (DEF KEY)

Eingabehilfen zeigen ein oder mehrere Zeichen an, wenn sie betätigt werden. Sofortausführungstasten *zeigen* einen Ausdruck, einen Befehl oder eine Anweisung *an und führen sie danach aus.*

Beispiel:
Definieren Sie die Taste [%] um, daß sie den Befehl LIST anzeigt und danach ausführt.

```
>def key '%','LIST'
```

Geben Sie das Zeichen % mit [SHIFT] [I/R] und dann [SHIFT] [5] ein. Das Semikolon wird *weggelassen.*

Speichern Sie die neue Definition mit [RTN] ab. Wenn Sie danach [SHIFT] [5] drücken, bedingt dies zwei Operationen:

1. Der String LIST wird kurzzeitig angezeigt.
2. Der Befehl LIST wird ausgeführt.

Das Ergebnis ist das gleiche, als wenn Sie LIST [RTN] eingeben würden. Beachten Sie die Bedeutung des Semikolons: das Semikolon bestimmt, ob der Anzeige der Tastenbelegung automatisch ein Wagenrücklauf/Zeilenvorschub folgt. Wenn ein DEF KEY Befehl mit einem Semikolon endet, wird der CR/LF unterdrückt, und die Tastenbelegung wird nur angezeigt. Wenn das Semikolon weggelassen wird, wird der String kurzzeitig angezeigt, bevor ein CR/LF gesendet und die Tastenbelegung ausgeführt wird.

Es ist wichtig, daß die Anzeigezeile gelöscht ist, bevor Sie eine Sofortausführungstaste drücken. Der HP-75 interpretiert die Anzeigezeile als ganzes, wenn daher in der Anzeige zusätzliche Zeichen stehen, können Sie Syntaxfehler erhalten.

Sie können alle Befehle und BASIC-Anweisungen, die Sie über das Tastenfeld ausführen können – wie zum Beispiel ASSIGN IO, PRINT und RUN – auch als Tastendefinitionen (in Groß- und Kleinschreibweise) eingeben.

Beispiel: Definieren Sie die Tastenkombination [CTL] [Z] so um, daß Sie damit das Programm SINGSONG aus Abschnitt 1 starten können.

```
>def key 'Ü','run "singsong"'
```

[CTL] [Z] erzeugt das Zeichen Ü. Beachten Sie, daß Sie den Filenamen SINGSONG in eigene Anführungszeichen einschließen müssen. Das Semikolon fällt weg.

```
>
```

Die Tastenfolge [CTL] [Z] ist neu belegt.

Mit CTL Z wird `run "singsong"` angezeigt, und, da Sie das Semikolon weggelassen haben, durch die CR/LF Sequenz gestartet. Sie hätten dasselbe erreicht, wenn Sie `run "singsong"` eingegeben und mit RTN abgeschlossen hätten: der HP-75 führt den Befehl `RUN` aus.

Abrufen von Tastendefinitionen (FETCH KEY)

Mit dem Befehl `FETCH KEY` können Sie Definitionen von Tasten oder Tastenkombinationen abrufen. Damit können Sie die Änderung von Tastendefinitionen verkürzen.

```
FETCH KEY 'Anzeigezeichen der Taste'
```

Beispiel: Prüfen Sie die momentane Belegung der nicht umgeschalteten Taste W . Drücken Sie FET und geben Sie ein:

```
>FETCH key 'w'
```

```
>DEF KEY 'w','w';
```

Momentan zeigt W das Zeichen `w` an. Der Cursor steht auf dem ersten Zeichen der Tastendefinition.

Mit dem Befehl `FETCH KEY` können Sie die Definitionen der 194 möglichen Tasten und Tastenkombinationen abrufen. Wenn Sie die derzeitige Definition einer umdefinierten Taste oder Tastenkombination anzeigen wollen, müssen Sie bei der Eingabe des Befehls `FETCH KEY` die ursprüngliche Anzeigefunktion der betreffenden Taste mit SHIFT I/R kurzzeitig wiederherstellen. Rufen Sie zum Beispiel die Definition von CTL Z, die eben geändert wurde, ab: drücken Sie FET und geben Sie `key '` SHIFT I/R CTL Z `'` ein.

Bevor Sie RTN drücken:

```
>FETCH key 'ü'
```

Mit SHIFT I/R und danach CTL Z erzeugen Sie ein `ü`.

Nach dem Drücken von RTN :

```
>DEF KEY 'ü','run "singsong"'
```

Die momentane Definition von CTL Z.

Eine Änderung der momentanen Definition einer Taste oder Tastenkombination erfordert drei Schritte:

1. Löschen Sie die Anzeige mit DEL oder SHIFT DEL. (Drücken Sie gegebenenfalls auch I/R für den Cursor.)
2. Geben Sie die neue Definition mit oder ohne Semikolon ein.
3. Drücken Sie am Ende der Operation RTN, oder brechen Sie den Vorgang mit ATTN CLR ab.

Löschen Sie die Tastendefinition mit SHIFT DEL, wenn Sie der Tastenkombination CTL Z wieder ihre ursprüngliche Funktion zuweisen wollen:

```
>DEF KEY 'ü','
```

Mit SHIFT DEL löschen Sie den Rest der Zeile ab der momentanen Cursorstellung.

Drücken Sie nochmals SHIFT I/R, dann CTL Z, und geben Sie den Rest der Definition ein:

```
>DEF KEY 'ü','ü';
```

Denken Sie an das Semikolon am Ende.

Nach RTN ist die ursprüngliche Anzeigefunktion von CTL Z wieder hergestellt.

Wenn Sie eine Tastendefinition mit dem Befehl `FETCH` abgerufen haben, können Sie jeden der beiden Parameter – die Tastendefinition oder das Anzeigezeichen der Taste – ändern.

Beachten Sie, daß der Befehl FETCH KEY *einfache* Anführungszeichen als äußere Begrenzung der Anzeigezeichen benutzt. Verwenden Sie deshalb *doppelte* Anführungszeichen, um einen Filenamen, Einheitscode usw. in der Tastendefinition zu kennzeichnen.

TIME- und APPT-Modi

Sie können mit dem Befehl DEF KEY Eingabehilfen und Sofortausführungstasten nicht nur für den EDIT-Modus, sondern ebenso auch für den TIME- und den APPT-Modus definieren.

Beispiel:

```
>def key 'z','set'
```

[Z] soll set anzeigen und ein CR/LF ausführen.

```
>
```

Wenn [Z] nun im TIME-Modus gedrückt wird, erscheint die Zeitsetzmaske.

Denken Sie daran, daß alle DEF KEY Befehle im EDIT-Modus eingegeben werden müssen.

Tastenmehrfachfunktionen (@)

Sie können zwei oder mehrere Anweisungen des EDIT-Modus in einem DEF KEY Befehl verknüpfen. Das Verknüpfungssymbol @ dient zum Verbinden von BASIC-Anweisungen und Befehlen des EDIT-Modus.

Beispiel: Definieren Sie die Tastenkombination [SHIFT] [=] so um, daß sie die folgenden vier Funktionen des HP-75 auslöst:

- Setzen einer Verzögerungsrate von 1 Sekunde pro Zeile (der Befehl DELAY).
- Setzen einer Anzeigelänge von 32 Spalten (der Befehl WIDTH).
- Anzeige des Katalogeintrags des momentanen Files (der Befehl CAT).
- Listen des Files (der Befehl LIST).

Dazu wird nur eine Tastendefinition benötigt:

```
>def key '|','delay1@width32@cat
@list'
```

Erzeugen Sie das Zeichen | mit [SHIFT] [=]. Zwischenräume und Groß/Kleinschreibung spielen in der Tastendefinition keine Rolle.

Nachdem Sie die Eingabe mit [RTN] abgeschlossen haben, führt die Tastenkombination [SHIFT] [=] nun die vier angegebenen Befehle aus.

Beachten Sie, daß Sie durch Weglassen der Zwischenräume eine Tastendefinition erzeugen können, die die Zeilenlänge des HP-75 überschreitet. Wenn eine solche lange Definition abgerufen wird, tritt Fehler 76 – line too long – auf, und nur die ersten 79 Zeichen der Tastendefinition werden in der DEF KEY Zeile angezeigt.

Erzeugung weiterer Tastenfiles (RENAME)

Im Speicher darf nur ein File mit dem Namens keys existieren. Solange sie untereinander *verschiedene* Filenamen haben, können jedoch beliebig viele Tastenfiles im Speicher vorhanden sein. Wenn Sie einen keys File im Moment nicht mehr benötigen, ihn aber für späteren Gebrauch aufheben wollen, können Sie ihm mit dem Befehl RENAME einen neuen Namen geben:

```
RENAME KEYS TO 'Filename'
```

Beachten Sie, daß der Name des aktiven Tastendefinitionfiles keys in allen Befehlen ohne Anführungszeichen erscheint.

Nach dem Umbenennen des keys-File steht der File mit seinem neuen Filenamen in Großbuchstaben im Systemkatalog. Alle zuvor neu belegten Tasten besitzen wieder ihre ursprüngliche Funktion. Nachfolgende Tastenbelegungen werden in einem neuen keys-File gespeichert, der vom HP-75 erzeugt und verwaltet wird. Wenn Sie einen früheren Tastenfile wieder aktivieren wollen, müssen Sie den momentanen keys-File erst löschen (PURGE) oder umbenennen (RENAME) und dann den folgenden Befehl ausführen:

```
RENAME 'Filename' TO KEYS
```

Der spezifizierte File wird zum aktiven keys File. Auf diese Weise können Sie von einem «maßgeschneiderten» Tastenfeld auf ein anderes umschalten.

Kopieren von Tastenfiles auf und von Massenspeicher (COPY)

Der keys File kann wie alle anderen Textfiles auf ein Massenspeichermedium kopiert und von dort eingelesen werden.

Kopieren Sie den keys File zum Beispiel auf Magnetkarten:

```
>copy keys to card
```

Lassen Sie die Anführungszeichen bei der Spezifikation des momentanen keys Files weg.

Entsprechend kann ein keys File von Magnetkarten in den Speicher eingelesen werden:

```
>copy card to keys
```

Da immer nur ein keys File im Speicher vorhanden sein darf, müssen Sie den momentanen keys File löschen oder umbenennen, bevor Sie die Operation COPY CARD TO KEYS durchführen können. Sofort, nachdem Sie einen keys File in den Speicher eingelesen haben, nehmen alle deklarierten Tasten und Tastenkombinationen sofort ihre neue Bedeutung an. Siehe auch Abschnitt 9, Seite 135.

Informationen für den fortgeschrittenen Benutzer

Der HP-75 verfügt über eine Reihe weiterer Kontrollmöglichkeiten zur Steuerung des Tastenfelds und der Anzeige:

- Sie können außer den Schreibmaschinentasten (wie Z) auch *Editiertasten* (wie TAB) und Systemtasten (wie SHIFT ATTN) neu belegen.
- Sie können mit neu definierten Tasten den Cursor steuern.
- Sie können den keys File direkt editieren.
- Sie können Tastendefinitionen und Anzeigezeichen mit Hilfe von Stringausdrücken spezifizieren.

Belegen von Editier- und Systemtasten (SHIFT I/R)

Jedem umgeschalteten, nicht umgeschalteten und Kontroll-Schreibmaschinenzeichen ist ein eindeutiges Anzeigezeichen zugeordnet. Das nicht umgeschaltete A erzeugt zum Beispiel ein a; SHIFT A erzeugt ein A und CTL A ein ▫. Auch die Editier- und Systemtasten besitzen entsprechende umgeschaltete, nicht umgeschaltete und Kontroll-Anzeigezeichen. Wenn Sie möchten, daß eine Editiertaste oder Systemtaste nicht ihre Funktion ausführt, sondern das ihr zugeordnete Zeichen anzeigt, drücken Sie einfach SHIFT I/R vor der Taste selbst. Drücken Sie zum Beispiel SHIFT I/R und dann TAB, um das der Taste TAB zugeordnete Zeichen anzuzeigen:

```
>τ
```

Die Taste TAB besitzt als Anzeigezeichen das unterstrichene kleine Tau.

Spezifizieren Sie Editier- und Systemtasten in DEF KEY Befehlen mit Hilfe ihrer Anzeigezeichen.

Beispiel: Ändern Sie die Funktion der Taste [TAB] so, daß sie drei Leerzeichen erzeugt.

```
>def key 'I','   ';
```

Erzeugen Sie wie zuvor mit [SHIFT] [I/R] das Anzeigezeichen von [TAB]. Geben Sie dann drei Leerzeichen für die Tastendefinition ein.

Speichern Sie diese Definition mit [RTN]. Wenn Sie jetzt [TAB] drücken, werden drei Leerstellen angezeigt.

Sobald Sie eine Editier- oder Systemtaste neu belegen, ist deren ursprüngliche Funktion verloren. Die Taste [TAB] tabuliert im TIME- und APPT-Modus nicht mehr, auch nicht, wenn Sie [SHIFT] [I/R] drücken. Sie müssen die ursprüngliche Funktion mit einem weiteren DEF KEY Befehl wieder herstellen:

```
>def key 'I','I',;
```

Beim Belegen müssen Sie zweimal [SHIFT] [I/R] drücken.

Sobald Sie [RTN] drücken, wird die Originalfunktion der Taste [TAB] wieder hergestellt. In Anhang D finden Sie eine Liste der den Editier- und Systemtasten zugeordneten Anzeigezeichen.

Belegen von [SHIFT] [ATTN]

Normalerweise schalten Sie den HP-75 mit der Tastenkombination [SHIFT] [ATTN] aus. Wenn Sie die Tastenkombination [SHIFT] [ATTN] neu definieren, wird die neue Tastenfunktion angezeigt oder ausgeführt, wann immer Sie [SHIFT] [ATTN] drücken, oder *wann immer der Computer versucht, sich auszuschalten*, nachdem die 5-minütige Ausschaltfrist abgelaufen ist.

Beispiel:

```
>def key '_','time$,date$'
```

Erzeugen Sie das Unterstreichungszeichen (_) mit [SHIFT] [I/R] und dann [SHIFT] [ATTN].

Nachdem Sie [RTN] gedrückt haben, zeigt [SHIFT] [ATTN] die Zeit und das Datum an. Die einzige Möglichkeit, den HP-75 auszuschalten, ist nun der Befehl BYE:

```
>bye
```

Schaltet den HP-75 aus.

Schalten Sie den HP-75 mit [ATTN] wieder ein. Bevor nicht [SHIFT] [ATTN] wieder auf die ursprüngliche Funktion zurückdefiniert ist, wird nach jedem Ausschaltintervall die Tastenfolge angezeigt und ausgeführt, und der HP-75 kann sich nicht ausschalten. Wenn Sie den Befehl RUN in eine Definition von [SHIFT] [ATTN] einbauen, kann der HP-75 immer dann ein Programm ausführen, wenn er sich normalerweise selbst ausschalten würde.

Cursorsteuerung

Mit Hilfe des Befehls DEF KEY, [SHIFT] [I/R] und der Editiertasten können Sie sowohl den Ersetzungscursor als auch den Einfügungscursor auf vielfältige Weise manipulieren. Spezifizieren Sie einfach die gewünschten Editieroperationen als Teil der Tastendefinition.

Beispiel: Definieren Sie die Tastenkombination [CTL] [1] so um, daß Sie damit den Cosinus einer beliebigen Zahl in der Anzeige berechnen können. Trigonometrischer Modus sei OPTION ANGLE RADIANS.

Bestimmen Sie zuerst die Tastenfolge, mit der Sie den Cosinus der folgenden Zahl in der Anzeige berechnen würden:

```
>4.5
```

Die Tastenfolge lautet:

1. Geben Sie die rechte Klammer ( ) ein.
2. Schieben Sie den Cursor an den Anfang der Anzeige ([SHIFT] [←]).
3. Setzen Sie den Einfügungscursor ([I/R]).
4. Geben Sie cos(ein.
5. Drücken Sie [RTN].

Wenn Sie die obenstehende Folge in eine DEF KEY Deklaration umwandeln wollen, müssen Sie mit [SHIFT] [I/R] drei der Tastenkombinationen der Folge erzeugen: [CTL] [1], [SHIFT] [←] und [I/R]. Die Folge, in eine DEF KEY Deklaration umgewandelt, wird zu:

def key '[SHIFT] [I/R] [CTL] [1]',') [SHIFT] [I/R] [SHIFT] [←] [SHIFT] [I/R] [I/R] cos('

Die Anzeige hat, bevor Sie [RTN] drücken, folgende Form:

```
>def key '1',')&~cos('
```

Fehlendes Semikolon bedingt sofortige Ausführung der Funktion.

[I/R] Anzeigezeichen.

[SHIFT] [←] Anzeigezeichen.

[CTL] [1] Anzeigezeichen.

Schließen Sie die Eingabe mit [RTN] ab. Sie können nun den Cosinus eines Arguments in der Anzeige durch das Drücken einer einzelnen Tastenfolge berechnen:

```
>4.5
```

Cosinus von 4.5 Radiant.

```
-.210795799431
```

Mit [CTL] [1] erhalten Sie das Ergebnis.

Praktisch alle Operationen zur Positionierung des Cursors und zur Editierung einer Zeile (wie zum Beispiel Löschen der Anzeige mit [CLR], Umschalten des Tastenfelds mit [SHIFT] [LOCK] und Abrufen des Anzeigebuffers mit [CTL] [FET]) können durch die Belegung von Tasten als Funktionen einer einzigen Taste erklärt werden.

Beispiel: Belegen Sie [CTL] [0] so, daß diese Tastenkombination die Anzeige löscht, FETCH KEY "" anzeigt, den Einfügungscursor setzt und nach links verschiebt, und die Anzeige der als nächstes gedrückten Taste vorbereitet.

```
>def key '0','FETCH KEY""~[(';
```

Schließen Sie die Eingabe mit [RTN] ab.

Semikolon.

[CTL] [0] [CLR] [I/R] [SHIFT] [I/R]

Rufen Sie die Definition einer beliebigen Taste, zum Beispiel [LOCK], ab. Drücken Sie zuerst [CTL] [0]:

```
>FETCH KEY "◆
```

Drücken Sie nun [LOCK]. Sie brauchen nicht zuerst [SHIFT] [I/R] zu drücken.

Dies ist die derzeitige Belegung von [CTL] [0].

```
>FETCH KEY "μ◆
```

Anzeige des Zeichens von [LOCK].

```
>DEF KEY 'μ','μ';
```

Mit [RTN] rufen Sie die derzeitige Definition von [LOCK] ab.

Hinweis: Die folgenden Tasten führen ihre normale Funktion aus und beenden dann die Ausführung der Definition, wenn sie in einer Tastendefinition spezifiziert werden und nicht mit [SHIFT] [I/R] eingeleitet werden: [TIME], [EDIT], [APPT], [FET], [↑], [↓], [RUN], [SHIFT] [ATTN]).

Editieren des keys-Files (EDIT KEYS)

Eine direkte Methode, Tastendefinitionen zu ändern, hinzuzufügen oder zu löschen ist das Editieren des keys-Files. Stellen Sie zuerst den Filepointer auf den keys-File. Benutzen Sie den Befehl EDIT KEYS.

```
>edit keys
```

Spezifiziert den keys-File. Der Filename steht nicht in Anführungszeichen.

```
keys      T   97 09:55 02/10/83
```

Der Katalogeintrag des keys- Files erscheint in der Anzeige.

Jetzt können Sie Ihre Tastendefinitionen direkt editieren. Holen Sie mit [↑] die erste Zeile des Files in die Anzeige:

```
:37 LIST
```

Die derzeitige Definition von [SHIFT] [5] – Listen des laufenden Files.

Ein Leerzeichen zwischen der Zeilennummer und der Tastendefinition zeigt, daß es sich um eine Sofortausführungstaste handelt.

Die Zeilennummer ist der Dezimalcode des mit [SHIFT] [5] assoziierten Anzeigezeichens (des Zeichens %).

Wenn zwischen der Zeilennummer und der Tastendefinition ein Semikolon (;) steht, ist die entsprechende Taste oder Tastenkombination eine Eingabehilfe. Ein Leerzeichen spezifiziert dagegen eine Sofortausführungstaste.

Sie können Belegungen von Tasten löschen, Tastendefinitionen editieren und neue Tastendefinitionen hinzufügen, indem Sie Zeilen im keys-File löschen, ändern oder hinzufügen. Die Tabelle mit dem Zeichensatz des HP-75 in Anhang D zeigt die einander entsprechenden Dezimalcodes, Anzeigezeichen und Tastenfolgen des HP-75. Beachten Sie, daß Tastendefinitionen im keys-File nur auftreten, wenn sie von der Originalbelegung der Taste abweichen.

Seien Sie vorsichtig, wenn Sie den keys-File editieren. Stellen Sie sicher, daß Ihre Zeilennummer dem Dezimalcode der gewünschten Taste entspricht. Wenn Sie einen Fehler machen (wie zum Beispiel einen RENUMBER Befehl ausführen), können Sie unerwünschte Belegungen erzeugen. Es ist zwar möglich, aber nicht empfehlenswert, die Tasten [EDIT], [ATTN] und [RTN] beim Editieren des keys-Files neu zu belegen. Wenn Sie zum Beispiel die Taste [EDIT] umdefinieren und dann [APPT] drücken, ist die einzige Möglichkeit, in den EDIT- Modus zurückzugelangen, einen Termin einzurichten, der im Befehlsfeld die Anweisung EDIT BASIC enthält.

Weiterhin kann nur die Taste [ATTN] bestimmte Systemfunktionen, wie zum Beispiel Unterbrechen der Programmausführung und Einschalten des HP-75, ausführen. Seien Sie auch vorsichtig, wenn Sie einen Tastenfile auf einem Peripheriegerät listen oder drucken lassen – die Einheit kann auf die Sonderzeichen im File auf unerwartete Weise reagieren.

Verlassen Sie den keys-File, indem Sie EDIT nochmals ausführen.

Stringausdrücke in DEF KEY Befehlen

Sie können in einer DEF KEY Deklaration Stringausdrücke als Parameter verwenden. Die allgemeinste Form des Befehls DEF KEY ist:

```
DEF KEY Einzeichen-Stringausdruck, Stringausdruck [;]
```

Ein Stringausdruck kann ein String in Anführungszeichen (wie alle bisherigen Beispiele), eine Stringvariable oder eine beliebige Stringkombination sein. Beachten Sie, daß der erste Parameter aus nur einem Zeichen bestehen darf.

Beispiel:

```
>def key chr$(26), 'print'
```

Mit [RTN] wird [CTL] [Z] (durch CHR$(26) spezifiziert) als Papiervorschubtaste für PRINTER IS Einheiten definiert.

Die Definition einer Taste kann auf ähnliche Weise abgerufen werden, indem Sie den Dezimalcode der Taste in einem FETCH KEY Befehl spezifizieren.

Beispiel:

```
>FETCH key chr$(26)
```

```
>DEF KEY 'ü','print'
```

Die Definition von [CTL] [Z].

Wenn die Tastendefinition mit FETCH KEY abgerufen wird, werden die beiden Parameter als Strings in Anführungszeichen angezeigt. Sie sehen, daß CHR$(26) in den String 'ü' umgewandelt wurde. Wenn Sie den Befehl FETCH KEY für spezielle Tasten wie [BACK] (Rückschritt) und [CTL] [J] (Zeilenvorschub) benutzen, können Sie ungewöhnliche Anzeigeinhalte erhalten, da diese Zeichen nicht wie andere Zeichen angezeigt werden. Es wird empfohlen, Spezialtasten in DEF KEY Deklarationen mit der Funktion CHR$ und Dezimalcodes zu spezifizieren.

Definition von Escapecodes (CHR$(27))

Escapecodes können als Tastendefinitionen spezifiziert werden. Erinnern Sie sich, daß Escapecodes Zeichenstrings sind, die mit dem Escapezeichen (Dezimalcode 27) beginnen, und die vom Anzeigefenster und von HP-IL Einheiten als Steuerinformation und nicht als Daten interpretiert werden.

Beispiel:

```
>def key 'W','print chr$(27);"&k
1S";'
```

[SHIFT] [W] wird umdefiniert und sendet jetzt einen Escapecode an PRINTER IS Einheiten.

Falls der Thermodrucker HP82162A als PRINTER IS Einheit deklariert wurde, wird er mit SHIFT W auf doppelten Zeilenabstand umgeschaltet.

Wenn ein Escapecode in einer Tastendefinition vom Anzeigefenster des HP-75 als Befehl erkannt wird, bewirkt die Tastenfolge eine sofortige Reaktion der Anzeige auf die Instruktion.

Beispiel: Die Tastenkombination CTL 2 soll so umdefiniert werden, daß sie den Cursor um eine Stelle nach rechts verschiebt. Der diese Cursorbewegung erzeugende Escapecode ist ESC C – die Taste muß also die beiden Zeichen CHR$(27) und C anzeigen.

```
>def key'2',chr$(27)&'C';
```

Das Anzeigefenster interpretiert die Tastendefinition als Steuerbefehl.

Jetzt dient CTL 2 nicht mehr als Eingabehilfe, sondern schiebt den Cursor um eine Position nach rechts. Sie finden eine Liste der Escapecodes, die zur Steuerung des Anzeigefensters des HP-75 und anderer HP-IL Anzeigeeinheiten benutzt werden können, in Anhang D, «Escapecodes zur Steuerung des Anzeigefensters».

Tastendefinitionen, die Eingaben erlauben (INPUT)

Sie können mit Hilfe der Anweisung INPUT Tastenfunktionen so definieren, daß sie in der Lage sind, bei ihrer Ausführung Eingaben von Daten anzunehmen.

Beispiel:

```
def key '4','input a$@copy card
to a$@edit a$'
```

Spezifiziert drei Operationen: INPUT, COPY und EDIT.

Nach der DEF KEY Deklaration hat die Tastenkombination CTL 4 folgende Funktionen:

- Annahme der Eingabe eines Filenamens, der in der Rechnervariablen A$ abgelegt wird.
- Start einer Kartenleseroperation, die einen Magnetkartenfile auf den angegebenen File im Speicher kopiert.
- Positionieren des Filepointers auf den Anfang des angegebenen Files.

Wenn Sie CTL 4 drücken, wird die Tastendefinition angezeigt, danach erfolgt ein Wagenrücklauf:

```
?
```

Geben Sie als Filenamen 'test' RTN ein.

```
Copy from card: Align & [RTN]
```

Der Computer ist bereit, einen Kartenfile als File TEST einzulesen.

Angenommen, es wurde tatsächlich ein File eingelesen, dann ist die letzte von CTL 4 ausgelöste Operation das Positionieren des Filepointers auf den Fileanfang:

```
TEST     B   213 12:04 02/10/83
```

Der Katalogeintrag des Files im Speicher.

Siehe auch Abschnitt 11, Seite 168 «Wertzuweisung über das Tastenfeld».

Teil III
Programmierung des HP-75

Abschnitt 11

Grundlagen der Programmierung

Inhalt

Einführung

Teil III dieses Handbuchs wurde unter der Annahme geschrieben, daß Sie schon einige Programmiererfahrung haben. In den folgenden Abschnitten werden Sie in die Syntax der BASIC-Anweisungen und Befehle des HP-75 eingeführt. Wenn Sie absoluter Anfänger in der Programmierung sind, ist es empfehlenswert, ein Buch über die Grundlagen der Programmierung in BASIC durchzuarbeiten, bevor Sie in diesem Handbuch weiterlesen. Vielleicht sind Sie ein erfahrener BASIC-Programmierer; in diesem Fall ist die Syntax-Zusammenfassung in Anhang H und das Referenzhandbuch des HP-75 alles, was Sie im Moment benötigen.

Dieser Abschnitt behandelt das Schreiben und Editieren von Programmen, die Steuerung der Programmausführung, die Benutzung von Programmvariablen und grundlegenden BASIC-Anweisungen sowie das Manipulieren von Files während der Programmausführung.

Schreiben und Editieren von Programmen

Beim Schreiben von BASIC-Programmen führen Sie die folgenden Schritte aus:

1. Wenn der momentane EDIT-File ein nichtleerer Arbeitsfile ist, müssen Sie ihn löschen (PURGE), benennen (NAME) oder umbenennen (RENAME).
2. Erzeugen Sie einen neuen BASIC-File mit Hilfe eines EDIT-Befehls.

Auf den folgenden Seiten wird ein Programm mit dem Namen KONTO entworfen; Sie sollten also edit 'konto',basic RTN eingeben. Der Katalogeintrag des neuen Files KONTO wird angezeigt; die Filelänge beträgt 0 Bytes.

3. Erzeugen Sie eine automatische Zeilennumerierung mit dem Befehl AUTO.
4. Geben Sie eine Programmanweisung nach der anderen mit Hilfe der Schreibmaschinen- und Editiertasten ein, und drücken Sie RTN, um vollständige Zeilen im File abzuspeichern.
5. Beenden Sie schließlich die Zeilennumerierung mit ATTN

Das Programm KONTO

Das Programm berechnet den Stand eines Kontos nach einer Lastschrift oder einer Gutschrift. Die untenstehende Listung zeigt das Programm KONTO, so wie Sie es vom Tastenfeld aus eingeben. Wenn Sie Programme lieber in Großbuchstaben eingeben, können Sie jetzt SHIFT LOCK drücken.

```
>10 ! Programm KONTO. K=Kontosta
nd; B$=Buchungsbetrag
>20 delay1@width32
>30 b$="0"!Initialisiert Buchung
sbetrag.
>40 input "Alter Kontostand: ";k
>50 input "Last- (-) oder Gutsch
rift (+): ",b$;b$
>60 b=val$(b$)!Umwandlung in num
erisch.
>70 k=k+b!Neuer Kontostand.
>80 disp "Neuer Kontostand: ";k
>90 wait 1
>100 if b#0 then 50
```

Beachten Sie, daß das Programm keine END Anweisung besitzt. Jeder BASIC-File wird mit einer unsichtbaren End-of-File Markierung in der letzten Zeile des Files erzeugt, deshalb ist die Anweisung END am Fileende nicht erforderlich. In den Beispielen dieses Handbuchs werden END Anweisungen am Fileende gewöhnlich weggelassen.

Drücken Sie RUN, um das Programm KONTO zu starten:

```
Alter Kontostand:
                    PRGM
```

Das Programm fragt nach dem alten Kontostand. Geben Sie 1000 RTN ein.

```
Last- (-) oder Gutschrift (+):0
                    PRGM
```

Das Programm fragt nach dem Betrag der Last- oder Gutschrift. Zahlen Sie DM 5.00 ein: 5 RTN.

```
Neuer Kontostand: 1005
                    PRGM
```

Der neue Kontostand wird 2 Sekunden lang angezeigt (Verzögerungszeit plus WAIT Intervall).

```
Last- (-) oder Gutschrift (+):5
                    PRGM
```

Danach wird der vorhergehende Betrag angezeigt. Geben Sie diesmal eine Einlage von DM 120.25 ein. Überschreiben Sie den angezeigten Betrag mit 120.25 und drücken Sie RTN.

```
Neuer Kontostand: 1125.25
                    PRGM
```

Wenn Sie die Programmausführung beenden wollen, löschen Sie zunächst den angezeigten Betrag mit [CLR] und geben danach 0 als Buchungsbetrag ein. Wenn Sie 0 eingeben, stoppt das Programm, die Statusanzeige **PRGM** wird zurückgesetzt, und der letzte Kontostand bleibt in der Anzeige stehen.

Erklärung des Programms. Der HP-75 führt Programmanweisungen in der Reihenfolge der Zeilennummern, beginnend bei Zeile 10, aus. Die erste Eingabeanweisung (INPUT, Zeile 40) erwartet den alten Kontostand. Die zweite INPUT Anweisung (Zeile 50) zeigt den Betrag der letzten Buchung an und erwartet die Eingabe eines neuen Betrags. Die Funktion VAL (Zeile 60) wandelt die Eingabe in einen Zahlenwert um. Die Anweisung IF (Zeile 100) schließlich prüft die Eingabe auf den Wert 0 und bestimmt, ob das Programm bei Zeile 50 fortgesetzt oder am Fileende zum Halten gebracht wird.

Starten von Programmen ([RUN], RUN)

Programme können im EDIT-Modus gestartet werden. Mit [RUN] starten Sie den momentanen BASIC-File; mit einer anderen Form des Befehls RUN können Sie *jedes beliebige* Programm im Speicher ausführen.

```
RUN  [Zeilennummer]
RUN  'Filename'  [Zeilennummer]
```

Beispiele:

```
>run
```

Startet den momentanen Programmfile bei der Zeile mit der niedrigsten Zeilennummer; gleiche Wirkung wie [RUN].

```
>run 30
```

Beginnt die Ausführung des momentanen Programms bei Zeile 30.

```
>run 'Hase'
```

Startet das Programm des angegebenen Files HASE bei der niedrigsten Zeilennummer.

```
>run 'Hase', 300
```

Beginnt die Ausführung bei Zeilennummer 300 im File HASE.

Wenn die spezifizierte Programmzeile nicht existiert, beginnt das Programm bei der nächsthöheren Zeilennummer.

Sie können auch während der Editierung von Files Programme ablaufen lassen; spezifizieren Sie einfach den Filenamen in einem RUN Befehl. Wenn Sie jedoch versuchen, einen Textfile zu starten, erhalten Sie Fehler 65 – access restricted. Bei dem Versuch, RUN KEYS oder RUN APPT auszuführen, erhalten Sie Fehler 84 – extra characters.

Wenn bei ausgeschaltetem HP-75 ein *Termin* fällig wird, schaltet sich der Computer in den EDIT-Modus. Falls der Termin den Befehl RUN enthält, wird das spezifizierte Programm ausgeführt. Nach der Beendigung des Programms schaltet sich der HP-75 wieder aus.

Initialisieren von Programmen

Der HP-75 initialisiert Programme, bevor er sie abarbeitet – das heißt, er stellt Speicherplatz für Programmvariablen zur Verfügung, setzt (initialisiert) die Variablen auf undefinierte Werte, prüft auf bestimmte Fehlerbedingungen (Seite 174), setzt andere Laufzeitbedingungen, und beginnt *dann* mit der Ausführung des Programms. Wenn lange Programme initialisiert werden, können Sie eine kleine Verzögerung bemerken.

Ein Programm wird nur in drei Fällen initialisiert: wenn Sie [RUN] drücken, einen RUN Befehl eingeben oder eine CALL Anweisung ausführen (Seite 230). Nach der Initialisierung zeigt der Katalogeintrag des Programms eine Vergrößerung der Filelänge, und die Funktion MEM zeigt eine entsprechende, aber größere Verminderung des verfügbaren Speicherplatzes. In Anhang D finden Sie die Speicherplatzanforderung von Programmen.

Unterbrechen von Programmen (ATTN, CONT)

Die Ausführung eines laufenden Programms wird durch das Drücken der Taste ATTN unterbrochen. Starten Sie zum Beispiel das Programm KONTO mit RUN:

```
Alter Kontostand:
                    PRGM
```

Drücken Sie jetzt ATTN, um die Programmausführung zu unterbrechen.

```
>
```

Das Programm bleibt initialisiert, auch wenn es unterbrochen wird.

Nach der Unterbrechung eines Programmes können Sie die meisten Tastenfeldoperationen – wie zum Beispiel Abfrage der Zeit, Berechnung eines numerischen Ausdrucks, Untersuchung oder Kopieren eines Files – durchführen, ohne daß der den Programmvariablen zugewiesene Speicherplatz freigegeben wird. (Bestimmte Befehle bewirken jedoch eine Deallokation aller unterbrochenen Programme, wenn eines davon in einen der Befehle COPY, DELETE, MERGE, NAME, PURGE, RENAME, RUN und TRANSFORM spezifiziert wird.) Die Ausführung eines allokatisierten Programmes kann mit dem Befehl CONT (*continue*) wieder da fortgesetzt werden, wo es unterbrochen wurde. (Wenn ein Programm bei einer Eingabeanforderung unterbrochen wurde, wird es mit dem Beginn der INPUT Anweisung fortgesetzt.)

```
CONT [Zeilennummer]
```

Beispiel:

```
>cont
```

```
Alter Kontostand:
                    PRGM
```

Die Ausführung wird fortgesetzt.

Sie können die Ausführung bei einer beliebigen Zeilennummer fortsetzen, wenn Sie diese Zeilennummer im Befehl CONT spezifizieren. Die Ausführung beginnt bei dieser oder der nächsten Zeile. Beachten Sie jedoch, daß Sie einen Fehler erhalten können, wenn Sie wichtige Programmzeilen überspringen oder wenn das Programm nicht mehr initialisiert ist.

Ein Unterschied zwischen RUN und CONT besteht darin, daß die Voreinstellung für den Zeilenparameter bei RUN der Programmanfang ist, während CONT auf die einer Unterbrechung folgende Zeile voreingestellt ist. Ein weiterer Unterschied ist, daß RUN jedes Programm im Speicher starten kann, während CONT nur den momentanen File anspricht. Ein dritter Unterschied besteht darin, daß RUN Programme initialisiert und vorhergehende Variablenwerte und Programmbedingungen zerstört, während CONT diese Zustände unverändert läßt.

Wenn Sie eine Programmzeile editieren - hinzufügen, ändern oder löschen – bedingt dies eine Deallokation des Programms. Die Ausführung eines CONT Befehls *vor* der Initialisierung oder *nach* der Deallokation eines Programms erzeugt Fehler 31 – CONT before RUN.

Beachten Sie, daß ein Programm initialisiert bleibt, auch wenn Sie ein anderes Programm editieren oder den HP-75 aus- und wieder einschalten. Eine einfache Methode zur Deallokation eines Programmes ist, eine nicht existierende Zeile aus dem Programm zu löschen oder ein anderes Programm zu starten.

Dies gilt auch für Programme, die mit der Anweisung STOP (Seite 165) und durch Fehlerbedingungen unterbrochen wurden (siehe Abschnitt 17).

Untersuchen von Programmen

Listen von Zeilen (LIST, PLIST)

Mit dem Befehl LIST können Sie einen ganzen File oder ausgewählte Zeilen eines Files auf der Anzeige des HP-75 und anderen DISPLAY IS Einheiten (Seite 128) listen lassen. Mit dem Befehl PLIST erfolgt die Listung auf den momentanen PRINTER IS Einheiten (das ist ebenfalls die Anzeige des HP-75, wenn Sie keinen HP-IL Drucker benutzen).

Beispiel:

```
>list
```

Listet den gesamten File. Unterbrechen Sie den Listvorgang mit [ATTN].

```
>plist 50
```

Listet Zeile 50 auf dem Drucker.

Der Befehl DELAY bestimmt die Geschwindigkeit des Listvorgangs im Anzeigefenster. Wenn Sie während einer Listung eine beliebige Taste außer [ATTN] drücken, wird sofort die nächste Zeile angezeigt. Die Befehle WIDTH und PWIDTH bestimmen die maximale Anzahl von Zeichen, die in jeder gelisteten Zeile auftreten. Bei einer WIDTH Spezifikation größer als 32 wandern längere Zeilen in der Anzeige (Seite 40). Die verbleibenden Programmlistings in diesem Handbuch sind alle mit einer WIDTH Einstellung von 32 dargestellt. Hier ist ein Listing des Programms KONTO:

```
10 ! Programm KONTO. K=Kontostan
d; B$=Buchungsbetrag
20 DELAY 1 @ WIDTH 32
30 B$="0" ! Initialisiert Buchun
gsbetrag.
40 INPUT "Alter Kontostand: ";K
50 INPUT "Last- (-) oder Gutschr
ift (+): ",B$;B$
60 B=VAL(B$) ! Umwandlung in num
erisch.
70 K=K+B ! Neuer Kontostand.
80 DISP "Neuer Kontostand: ";K
90 WAIT 1
100 IF B#0 THEN 50
```

Beachten Sie verschiedene Merkmale des Listings:

- Die Zwischenräume wurden zur besseren Lesbarkeit geändert.
- Die Schlüsselworte und Variablennamen wurden in Großbuchstaben umgewandelt; nur Programmkommentare und Zeichen in Anführungszeichen blieben in ihrer Schriftart und Struktur erhalten.
- Die Eingabeaufforderung und der Cursor erscheinen nicht. Die Zeilen können nur gelesen, nicht aber editiert werden. Wenn Sie nach einem Listing eine Taste drücken, erscheinen wieder der Cursor und die Eingabeaufforderung.

Filepointer und Programmpointer ([FET], [↑], [↓])

Erinnern Sie sich an Abschnitt 3: der HP-75 markiert die anstehende Zeile – die momentan direkt editierbare Zeile – mit dem *Filepointer*. Drücken Sie [FET] und dann [RTN], um die anstehende Zeile abzurufen oder anzuzeigen:

```
>40 INPUT "Alter Kontostand: ";K
```

Der Filepointer zeigt auf die Zeile, bei der das Programm zuletzt mit [ATTN] unterbrochen wurde.

Die Eingabeaufforderung und der Cursor zeigen an, daß die Zeile editierbereit ist. Sie können die Zeile mit [CLR], [ATTN] oder [EDIT] aus der Anzeige löschen. Die Position des Filepointers ändert sich jedoch erst bei Ausführung einer der folgenden Operationen:

- Die Ausführung des gleichen Programms wird bei einer anderen Zeile mit der Taste [ATTN], durch einen Fehler oder durch einen Befehl STOP oder END (Seite 165) angehalten. Wenn das Programm zum Beispiel durch einen Fehler angehalten wurde, können Sie die fehlerhafte Zeile mit [FET] und dann [RTN] zur Editierung in die Anzeige rufen.
- Der Filepointer wird mit [↑], [↓], [SHIFT] [↑] oder [SHIFT] [↓] bewegt.
- Eine andere Zeile des Files wird mit FETCH abgerufen.
- Ein anderer File wird editiert (EDIT oder NAME).
- Der momentane File wird gelöscht (PURGE).

Die Programmierung der Editierbefehle des HP-75 wird später in diesem Abschnitt besprochen.

Mit dem *Programmpointer* markiert der HP-75 die nächste auszuführende Anweisung. Der Programmpointer wird unabhängig vom Filepointer gesetzt. Wenn Sie die Stellung des Filepointers ändern, wird die Stellung des Programmpointers damit nicht geändert; der Filepointer hat also keine Auswirkung auf die Reihenfolge der Programmausführung.

Beispiel:

1. Starten Sie das momentane Programm mit [RUN].
2. Unterbrechen Sie das Programm mit [ATTN].
3. Prüfen Sie die Stellung des Filepointers mit [FET] und [RTN].
4. Stellen Sie den Filepointer mit [SHIFT] [↓] auf die letzte Zeile des Files.
5. Geben Sie [CLR] cont [RTN] ein. Das Programm wird von dort fortgesetzt, wo es unterbrochen wurde, und *nicht* von der letzten Zeile des Files.

Ein «normal» beendetes Programm – durch die Anweisung END oder durch die End-of-File Markierung – beeinflußt nicht die Stellung des Filepointers.

Beispiel:

1. Unterbrechen Sie das momentane Programm mit [ATTN].
2. Stellen Sie den Filepointer mit [SHIFT] [↑] auf die erste Zeile des Programms.
3. Starten Sie das Programm erneut mit [RUN].
4. Geben Sie als Buchungsbetrag eine 0 ein, um die Programmausführung zu beenden.
5. Zeigen Sie mit [FET] und dann [RTN] die anstehende Zeile an – es ist noch immer die erste Zeile des Files.

Wenn ein Programm unterbrochen wird, zeigt der Filepointer also auf die Stellung des Programmpointers. Wird das Programm normal beendet, bleibt die Stellung des Filepointers unverändert.

Abrufen von Zeilen (FETCH)

Sie können mit dem Befehl FETCH bestimmte Programmmzeilen zur Editierung in die Anzeige holen.

Beispiele:

```
>FETCH 20
```

Spezifiziert eine Zeilennummer.

```
>20 DELAY 1 @ WIDTH 32
```

Holt Zeile 20.

```
>FETCH 'K+B'
```

Spezifiziert einen Suchstring.

```
>70 K=K+B ! Neuer Kontostand.
```

Holt die nächste Zeile *nach* der anstehenden Zeile, die den Suchstring enthält.

```
>FETCH 'P',0
```

Spezifiziert einen Suchstring und eine Zeilennummer.

```
>10 ! Programm KONTO. K=Kontosta
```

Holt die nächste Zeile (nach der spezifizierten Zeile), die den Suchstring enthält.

Der *Suchstring* muß genauso eingegeben werden, wie er im Programm auftritt – gewöhnlich in Großbuchstaben – oder es kann keine Übereinstimmung gefunden werden.

Definitionen

Jede Zeile eines BASIC-Programms enthält folgende Informationen:

- Eine *Zeilennummer*, eine vorzeichenlose ganze Zahl zwischen 0 und 9999, die die Reihenfolge der Programmausführung bestimmt.
- Ein *Schlüsselwort*, wie zum Beispiel `INPUT`, das die Kernanweisung der Zeile darstellt. Schlüsselworte sind das Vokabular, das der HP-75 erkennt und in den internen Maschinencode übersetzt.
- Null oder mehr *Parameter*, die Informationen für die Anweisung des Schlüsselwortes liefern.

Beispiel:

```
100 BEEP 440,.5
```

Eine Programmzeile mit Zeilennummer, Schlüsselwort und Parametern.

Ein Schlüsselwort mit Parametern ist entweder eine BASIC-*Anweisung* oder ein *Systembefehl*. Im normalen Sprachgebrauch bedeutet Anweisung generell eine Instruktion, die innerhalb eines Programmes den Programmablauf steuern, und Befehl bedeutet eine Instruktion, die *direkt* auf ein Programm oder auf das System wirken. Als Anweisungen werden gewöhnlich in ein Programm eingegebene Zeilen, als Befehle direkt vom Tastenfeld aus eingegebene Instruktionen bezeichnet. Beim HP-75 ist die Trennung zwischen *Anweisung* und *Befehl* zu einem gewissen Grad zufällig, da die meisten Anweisungen auch über das Tastenfeld eingegeben werden und die meisten Befehle auch programmiert werden können. Trotzdem ist es manchmal nützlich, diese Unterscheidung zu treffen, und wir werden dies auch tun, wo es angebracht erscheint.

Die folgenden BASIC-Anweisungen können nicht über das Tastenfeld ausgeführt werden:

`END` `STOP`	In diesem Abschnitt besprochen.

`POP`
`GOTO`
`FOR*`
`NEXT*`
`GOSUB`
`RETURN`
`ON TIMER`
`OFF TIMER`
	In Abschnitt 12 besprochen.

`DEF FN`
`LET FN`
`END DEF`
`OPTION BASE`
	In Abschnitt 13 besprochen.

`READ`
`RESTORE`
`DATA`
	In Abschnitt 14 besprochen.

`IMAGE`	In Abschnitt 16 besprochen.
`ON ERROR` `OFF ERROR`	In Abschnitt 17 besprochen.

* Sie können eine `FOR...NEXT` Schleife über das Tastenfeld ausführen, wenn alle Anweisungen der Schleife mit @-Symbolen verknüpft sind.

Die folgenden Systembefehle sind nicht programmierbar:

```
ADJUST
EXACT
RESET
SET
STATS
```

} In Abschnitt 6 besprochen.

Die folgenden Systembefehle setzen einen Rechnerzustand:

```
ALARM OFF
ALARM ON
ASSIGN IO
BEEP OFF
BEEP ON
DEFAULT OFF
DEFAULT ON
DEF KEY
DELAY
DISPLAY IS
ENDLINE
LOCK
MARGIN
OFF IO
OPTION ANGLE DEGREES
OPTION ANGLE RADIANS
PRINTER IS
PWIDTH
RESTORE IO
STANDBY OFF
STANDBY ON
TRACE FLOW
TRACE OFF
TRACE VARS
WIDTH
```

Nachdem ein Rechnerzustand gesetzt ist, sei es über das Tastenfeld oder von einem Programm, bleibt er wirksam, bis er mit einem neuen Befehl geändert wird. Im Anhang D finden Sie die Voreinstellungen nach einem Zurücksetzen des Systems.

Zeilen mit Mehrfachanweisungen

Zeilen mit *Mehrfachanweisungen* sind Programmzeilen, die zwei oder mehr Anweisungen oder Befehle enthalten. Der HP-75 erkennt das Zeichen @ als Verknüpfungssymbol.

Beispiel:

```
230 DISP A @ BEEP @ RETURN
```

Verbinden dreier Programm-Anweisungen mit dem Symbol @.

Mehrfachanweisungen bieten zwei Vorteile:

- Sie sparen Speicherplatz (siehe Anhang D).
- Die Ausführungsgeschwindigkeit wird geringfügig erhöht.

Sie sollten bei Mehrfachanweisungen verschiedene Dinge beachten:

- Es ist möglich, Zeilen mit mehr als 96 Zeichen einzugeben, wenn Sie abgekürzte Schlüsselworte und eine komprimierte Schreibweise benutzen. Beim Listen oder Abrufen erzeugen lange Zeilen Warnung 67 – `line too long` – und nur die ersten 94 Zeichen der Zeile werden angezeigt. Der Rest der Zeile bleibt im File erhalten, Sie müssen die Zeile zum Editieren aber verkürzen. Wenn Sie eine lange Zeile in der Anzeige haben und RTN drücken, versucht der HP-75, nur die ersten 94 Zeichen der Zeile als vollständige Programmzeile zu interpretieren.

- Die Anweisungen DATA, DEF FN und IMAGE sollten in Mehrfachanweisungen nicht verwendet werden.
- Die folgenden Anweisungen sollten immer als letzte Anweisung einer Zeile erscheinen: DIM, INTEGER, REAL, SHORT, GOTO, END, RETURN und END DEF. Wenn in einer Zeile mit Mehrfachanweisungen eine GOTO Anweisung steht, wird verzweigt, bevor die nächste Anweisung in der Zeile ausgeführt werden kann.
- Wenn Sie Vergleichsanweisungen (IF . . . THEN . . . ELSE) verbinden, sollten Sie bedenken, was das Ergebnis des Booleschen Vergleichs sein wird. Wenn die Bedingung erfüllt ist, werden alle nach THEN verknüpften Instruktionen ausgeführt; andernfalls werden alle verknüpften Instruktionen nach ELSE ausgeführt.

Programmvariablen

Sie haben drei Arten von Programmvariablen zur Verfügung:

- *Einfache numerische* Variablen repräsentieren einzelne Zahlenwerte (siehe Abschnitt 5).
- *Numerische Feld*variablen speichern mehrere Zahlenwerte (siehe Abschnitt 13).
- *String*variablen speichern Zeicheninformation. Diese werden im folgenden besprochen.

Die Wertzuweisung auf Rechner- und Programmvariablen erfolgt mit Hilfe von LET, [=], INPUT und READ$, an Programmvariablen zusätzlich mit READ und als Parameter in der Anweisung DEF FN.

Stringvariablen

Namen von Stringvariablen können aus beliebigen Buchstaben oder Buchstabenkombinationen, gefolgt von einem Dollarzeichen ($), bestehen. Gültige Namen für Stringvariablen sind zum Beispiel a$, C1$, X9$ und Y$.

Der HP-75 geht davon aus, daß eine Stringvariable weniger als 33 Zeichen speichert, wenn Sie die Maximallänge nicht mit einer DIM-Anweisung ändern (Seite 194). Die Zuweisung von Zeichen auf eine Stringvariable erfolgt mit Hilfe ihres Namens und des Gleichheitszeichens.

Beispiel:

```
>a$='hallo'
```

Mit [RTN] weisen Sie A$ die fünf Zeichen in Anführungszeichen zu.

Sie können einer Stringvariablen beliebige Zeichenkombinationen, einschließlich Kommata, Leerzeichen und Kontrollzeichen, nicht aber die Anführungszeichen, mit denen Sie den String kennzeichnen, zuweisen.

Die Werte von Rechnerstringvariablen bleiben erhalten, bis ihnen neue Werte zugewiesen werden, oder bis Sie den Befehl CLEAR VARS ausführen (Seite 81).

Verketten von Strings (&)

Bei String*verkettungen* wird ein String an das Ende eines anderen Strings angehängt. Zur Verkettung zweier Strings wird das Zeichen & benutzt.

Beispiele:

```
>'ganz'&'zu'&'sammen'
```

Verkettet drei Strings mit &.

```
ganzzusammen
```

Das Ergebnis. Beachten Sie, daß die Strings ohne führende oder nachlaufende Leerzeichen angezeigt werden.

```
>a$='super'@b$='mann'
c$=a$&b$
```

Die Verkettung von A$ und B wird C$ zugewiesen.

```
>c$
```

```
supermann
```

Anzeige des Werts von C$.

Die beiden verketteten Strings.

Die *Stringfunktionen* des HP-75 werden in Abschnitt 13 behandelt.

Wertzuweisung auf Programmvariablen (LET, [=])

Häufig werden das Gleichheitszeichen und die Anweisung LET zur Wertzuweisung auf Variablen verwendet. LET ist dabei optional.

```
[LET] einfache Variable [, einfache Variable...]=numerischer Ausdruck
[LET] Stringvariable [, Stringvariable...]=Stringausdruck
```

Die folgenden Anweisungen sind alle gleichwertig:

```
100 X=12        100 X=3*4        100 LET X=12    100 LET X=3*4
```

Auch Wertzuweisungen auf mehrere Variablen sind möglich:

```
30 X,Y,Z=0
40 A$,B$,T$=''
```

Wenn in einer Berechnung eine numerische Variable benutzt wird, der noch kein Wert zugewiesen wurde, erhalten Sie Warnung 7 – no value –, und die Berechnung wird mit dem Ersatzwert 0 fortgesetzt. Wenn entsprechend ein String benutzt wird, bevor ihm ein Wert zugewiesen wurde, wird no value gemeldet, und der Nullstring wird als Ersatzwert benutzt. Diese Warnbedingung kann in eine Fehlerbedingung geändert werden (kein Ersatzwert; die Ausführung des Programms wird unterbrochen), wenn Sie den Ersatzwert-Modus des HP-75 desaktivieren (default off [RTN]).

Bei einer Stringzuweisung darf der String nicht mehr Zeichen enthalten als die Variablenlänge erlaubt; andernfalls erhalten Sie Fehler 42 – string too long.

Im allgemeinen ist es nützlich, Variablen zu Beginn von Programmen und Unterprogrammen zu initialisieren, das heißt, ihnen Anfangswerte zuzuweisen.

Wichtige Anweisungen

Anhalten der Programmausführung (END, STOP)

Zwei Anweisungen halten die Ausführung eines Programms an:

```
END
```

```
STOP
```

Nach STOP können Sie Werte von Variablen prüfen und ändern, die meisten anderen Tastenfeldoperationen durchführen, und danach die Programmausführung mit dem Befehl CONT wieder fortsetzen. Wenn in einem laufenden Programm eine END-Anweisung ausgeführt wird, bedingt dies eine Freigabe des den Programmvariablen zugewiesenen Speicherinhalts (eine Deallokation des Programms); deren Werte sind damit verloren.

Wenn Sie vom Tastenfeld aus [ATTN] drücken, hat dies die gleiche Wirkung wie die Anweisung STOP in einem Programm. Das Ende eines BASIC-Files hat die gleiche Wirkung wie die Anweisung END in einem Programm.

Suspendieren der Ausführung (WAIT)

Mit einer WAIT-Anweisung halten Sie den HP-75 für eine bestimmte Zeit an, bevor die Programmausführung automatisch fortgesetzt wird.

```
WAIT Anzahl Sekunden
```

Die *Anzahl Sekunden* kann durch einen beliebigen numerischen Ausdruck spezifiziert werden. Der Wert wird vom HP-75 auf 0 bis 2^{26} Sekunden (ungefähr 2 Jahre) begrenzt, die Auflösung beträgt 0.1 Sekunden. Negative Parameter erzeugen eine Wartezeit von 0 Sekunden.

Wenn die Anweisung WAIT einer DISP-Anweisung folgt, wird die Zeile, die mit DISP angezeigt wird, zusätzlich für die in WAIT spezifizierte Zeit angezeigt. Die Ausführung der nächsten Anweisung wird um das gesamte Zeitintervall verzögert.

Mit [ATTN] können Sie eine WAIT-Anweisung aufheben und das Programm anhalten.

Programmkommentare (REM, !)

Gelegentlich ist es sinnvoll, zur internen Dokumentation eines Programms Kommentare einzufügen. Dies kann mit Hilfe der Anweisung REM (*remark*) oder des Kommentarzeichens ! erfolgen.

```
REM [beliebige Zeichenkombination]
!   [beliebige Zeichenkombination]
```

Das Schlüsselwort REM darf nur als erstes Wort einer Programmzeile benutzt werden; das Kommentarzeichen ! kann an beliebiger Stelle in einer Programmzeile nach der Zeilennummer stehen. ! sollte in den Anweisungen IMAGE oder DATA nicht als Kommentarzeichen verwendet werden. Beachten Sie, daß *alle* Zeichen, die nach einem ! oder REM folgen, als Teil eines Kommentars betrachtet werden; Kommentare sollten deshalb immer am Ende einer Zeile stehen.

Anzeigen und Drucken von Information (DISP, PRINT, TAB)

Die Anweisung DISP. Mit der Anweisung DISP (*display*) können Sie numerische und Stringinformation entsprechend den derzeitigen Werten von DELAY und WIDTH auf der Anzeige des HP-75 und auf anderen DISPLAY IS Einheiten ausgeben.

```
DISP [Anzeigeliste]
```

Die Anzeigeliste kann Zeichen in Anführungszeichen, Variablennamen, numerische und Stringausdrücke und die Funktion TAB (Seite 167) enthalten, die mit Kommata oder Semikolons getrennt (und möglicherweise abgeschlossen) werden.

Wenn Sie keine Parameter spezifizieren, erzeugt die Anweisung DISP eine leere Zeile.

Zahlen werden mit einer führenden Leerstelle oder einem Minuszeichen und mit einer nachlaufenden Leerstelle dargestellt. Zeichenstrings werden ohne führende oder nachlaufende Leerzeichen angezeigt.

Semikolons und Kommata unterdrücken den Wagenrücklauf/Zeilenvorschub nach der Anzeige von Ausgabeelementen. Bei einem Semikolon wird das nächste Element direkt an das vorhergehende angeschlossen; bei einem Komma wird das vorhergehende Element linksbündig in einem 21-stelligen Anzeigefeld angezeigt. Die nachfolgende Information wird also von der Stelle an angezeigt, die sich 21 Spaltenpositionen rechts vom Beginn des vorhergehenden Elements befindet. Wenn die Zeilenlänge nicht ausreicht, wird das nachfolgende Element in der nächsten Zeile angezeigt.

Beispiel: Das folgende Programm zeigt 13 Zeichen in der gleichen Zeile an, dabei führt es die Anweisung DISP 13mal aus.

```
10 FOR I=14 TO 26
20 DISP CHR$(I);
30 NEXT I
```

Zeigt die Zeichen mit den Dezimalcodes 14 bis 26 an.

Falls der Wagenrücklauf/Zeilenvorschub am Ende der Programmausführung unterdrückt wird, erscheinen der Cursor und die Eingabeaufforderung beim Drücken von [CLR] *nach* dem letzten Element in der Anzeige. Sie können die Ausgabeoperation mit einer weiteren Anweisung DISP (zum Beispiel 40 DISP) vervollständigen; danach erscheint der Cursor linksbündig.

Gewöhnlich ist es *nicht notwendig, das Schlüsselwort* DISP *selbst einzugeben.* Geben Sie einfach die Anzeigeelemente ein und drücken Sie [RTN]. Die Anweisung DISP wird dann mit ihren Parametern in den BASIC-File eingetragen.

Beispiel:

Zeilen, die so eingegeben wurden...

```
10 "Die Zeit ist reif!"
```

```
20 a;b;100
```

```
30 chr$(i);
```

...erscheinen so...

```
10 DISP "Die Zeit ist reif!"
```

```
20 DISP A;B;100
```

```
30 DISP CHR$(I);
```

Bei Mehrfachanweisungen muß die implizite DISP-Anweisung immer die erste Anweisung in der Zeile sein.

Die Anweisung PRINT. Mit der Anweisung PRINT werden Informationen auf PRINTER IS Einheiten ausgedruckt.

```
PRINT [Druckliste]
```

Die *Druckliste* kann wie die *Anzeigeliste* Zeichen in Anführungszeichen, Variablennamen, numerische und String-ausdrücke und die Funktion TAB, durch Kommata oder Semikolons getrennt (und möglicherweise beendet), enthalten.

Wenn dem Schlüsselwort PRINT keine Parameter folgen, führen die Druckereinheiten eine Zeilenschaltung durch (sie geben eine leere Zeile aus). Die Zeilenschaltung wird unterdrückt, wenn eine PRINT Anweisung mit einem Semikolon endet.

Die Funktion TAB. Sie können die Funktion TAB zusammen mit DISP und PRINT verwenden, um Informationen an bestimmten Spaltenpositionen auszudrucken. TAB springt zur spezifizierten Spalte, bevor der nächste Parameter angezeigt oder gedruckt wird.

```
TAB (Spaltenposition)
```

Sie können mit einem beliebigen numerischen Ausdruck einen positiven Wert für die Spaltenposition spezifizieren. Dezimalzahlen werden auf ganzzahlige Werte gerundet. Wenn eine in TAB spezifizierte Spaltenposition außerhalb des von WIDTH oder PWIDTH gestatteten Wertes liegt, wird diese Position auf einen Wert innerhalb des erlaubten Bereichs reduziert. Ein negatives Argument oder Null erzeugen Warnung 54 – invalid TAB – und die Funktion TAB nimmt den Ersatzwert 1 an.

Wenn Sie TAB benutzen, haben Sie die Zeilenbegrenzung durch die derzeitige WIDTH oder PWIDTH zu berücksichtigen. In den folgenden Beispielen wird eine Einstellung von WIDTH von 32 oder größer angenommen:

```
>print 'Hier';tab(18);'Jetzt'
```

```
Hier          Jetzt
```

Positionierung des zweiten Zeichenstrings (JETZT).

Das erste Zeichen von Jetzt wird in Spalte 18 gedruckt.

```
>tab(30);-2
```

```
                              -2
```

Das Schlüsselwort DISP ist nicht erforderlich.

Das Minuszeichen der Zahl wird in Spalte 30 angezeigt.

Nach der Funktion TAB sollten Sie ein Komma, und nicht ein Semikolon, verwenden, damit der nächste Parameter bei der spezifizierten Spalte beginnend angezeigt oder ausgedruckt wird.

Beachten Sie, daß Sie mit TAB den Cursor nicht zurücksetzen können. Die Angabe eines Spaltenparameters kleiner als die momentane Spaltenposition bewirkt die Ausgabe des nächsten Elements in der spezifizierten Spalte auf der Folgezeile.

Weitere Information zum Anzeigen und Drucken von Daten finden Sie in Abschnitt 16, «Anzeige- und Druckformatierung».

Wertzuweisung über das Tastenfeld (INPUT)

Mit der Anweisung INPUT kann ein Programm die Zuweisung von Zahlen und Zeichen über das Tastenfeld auf Variablen anfordern. INPUT erlaubt die Kommunikation mit einem laufenden Programm. Die Anweisung INPUT kann drei Formen annehmen:

```
INPUT Variable [, Variable...]
INPUT "Eingabeaufforderung"; Variable [, Variable...]
INPUT "Eingabeaufforderung", Eingabeaufforderungsstring; Variable [, Variable...]
```

Wenn die Anweisung INPUT nur aus einer Variablenliste besteht, erscheint bei der Ausführung der Anweisung an der momentanen Position ein Fragezeichen als Eingabeaufforderung:

Beispiel:

```
100 INPUT N$
```

Die Anweisung spezifiziert keine Eingabeaufforderung.

Wird so ausgeführt:

```
?
```

Fordert mit einem Fragezeichen zur Eingabe auf. Sie können einen String mit oder ohne Anführungszeichen eingeben.

Sie können eine beliebige Kombination von numerischen und Stringvariablen als INPUT Liste spezifizieren.

Beispiel:

```
60 INPUT X,Y,Z
```

Spezifiziert drei einfache numerische Variablen.

Wenn eine INPUT-Anweisung mehr als eine Variable anfordert, geben Sie deren Werte durch Kommata getrennt in die gleiche Zeile ein:

```
?7.5,10,13.7
                    PRGM
```

Drücken Sie danach RTN.

INPUT-Variablen werden von links nach rechts mit dem Tastenfeldwert belegt. Die Eingaben für numerische Variablen müssen *Zahlen* sein; numerische Ausdrücke (zum Beispiel SIN(X)) sind nicht möglich. Die Eingaben für Stringvariablen werden als reiner Text, das heißt, nur als Zeichen behandelt. Die Eingaben für Stringvariablen können Zeichen mit oder ohne Anführungszeichen sein, aber eine Eingabe ohne Anführungszeichen sollte kein Komma enthalten (da Kommata Eingabeteile beenden). Auch führende und nachlaufende Leerzeichen, falls sie nicht zum String gehören, werden Stringvariablen nicht zugewiesen. Wenn Sie bei der Eingabe einen Fehler machen, erhalten Sie eine Fehlermeldung, und die Eingabe wird nochmals angefordert.

Wenn Sie in einer INPUT-Anweisung eine *Eingabeaufforderung in Anführungszeichen* spezifizieren, erscheint dieser String anstelle des Fragezeichens.

Beispiel:

```
100 INPUT "Name=";N$
```

Spezifiziert die Eingabeaufforderung.

Bei der Ausführung:

```
Name=
                    PRGM
```

Die Zeichen der Eingabeaufforderung sind *geschützt*; sie können nicht überschrieben werden.

Sie können auch vor INPUT eine Anweisung DISP einfügen, um einen String als Eingabeaufforderung zu erhalten.

Beispiel:

```
100 DISP 'Name=';
110 INPUT N$
```

Die Anweisung DISP wird mit einem Semikolon abgeschlossen, um CR/LF zu unterdrücken.

```
Name=?
                    PRGM
```

Die vorgegebene Eingabeaufforderung ? erscheint in der gleichen Zeile wie der angezeigte String.

Wenn Sie bei einer INPUT-Anweisung zusätzlich zu der Eingabeaufforderung in Anführungszeichen einen *Eingabeaufforderungsstring* spezifizieren, wird der momentane Wert dieses Stringausdrucks bei der Ausführung von INPUT angezeigt, und der Cursor steht auf dem ersten Zeichen des Strings.

Beispiel:

```
100 INPUT 'Name=',N$;N$
```

Die erste Eingabeaufforderung muß ein String in Anführungszeichen sein; die zweite Eingabeaufforderung kann ein beliebiger Stringausdruck sein.

Angenommen, N$ hat zur Zeit den Wert Pia, dann wird die Anweisung so ausgeführt:

```
Name=Pia
                    PRGM
```

Die zweite Eingabeaufforderung kann editiert werden.

Mit [CLR] können Sie das die Zeile ab der momentanen Cursorposition, nicht aber die geschützten Zeichen der Eingabeaufforderung in Anführungszeichen, löschen. Sie können den Eingabeaufforderungsstring überschreiben oder einfach wieder eingeben. Mit [RTN] wird der Wert der Variablen in N$, der ersten und in diesem Fall einzigen Variablen der Eingabeliste, gespeichert. Mit der erweiterten Anweisung INPUT, die zuvor schon im Programm KONTO verwendet wurde, können Sie auf einfache Weise Werte von Variablen abrufen und erneuern.

Wenn Sie vor dem Eingabeaufforderungsstring keine weiteren Zeichen anzeigen wollen, können Sie als Eingabeaufforderung in Anführungszeichen einfach den Nullstring (' ' oder " ") spezifizieren.

Beispiel:

```
100 INPUT '',N$;N$
```

Es wird kein Zeichen vor dem Wert von N$ angezeigt.

Sie können auch verschiedene Variablen als Eingabevariablen und als Eingabeaufforderungsvariablen spezifizieren:

Beispiel:

```
100 INPUT 'Name=',N$;P$
```

Der momentane Wert von N$ wird angezeigt; der Eingabewert wird aber auf P$ abgespeichert.

Insgesamt können Sie als Antwort auf eine INPUT-Anweisung maximal 94 Zeichen eingeben, *minus* die Anzahl der Zeichen der Eingabeaufforderung in Anführungszeichen. Wenn die Länge der angezeigten Zeile 32 Zeichen überschreitet, wandern die Zeichen bei der Eingabe nach links. Danach können Sie den Cursor mit [←] und [→] auf alle Zeichen (außer auf die Eingabeaufforderung in Anführungszeichen) positionieren. Sie können die ersten 32 Zeichen betrachten, indem Sie [SHIFT] [←] drücken und [←] gedrückt halten; sobald Sie [←] loslassen, erscheint der Cursor eingabebereit am Ende der Eingabeaufforderung in Anführungszeichen. (Wenn die Eingabeaufforderung mehr als 64 Zeichen enthält, können die mittleren Zeichen nicht mehr betrachtet werden, nachdem sie über die Anzeige gewandert sind.)

Der Befehl MARGIN (Seite 40) setzt die Anzahl Zeichen, die bei INPUT eingegeben werden können, bevor das Zeilenende akustisch angekündigt wird.

Beachten Sie, daß Sie eine Eingabe mit [RTN], [RUN], [↑], [↓] oder [FET] abschließen können. Im Programm KONTO zum Beispiel können Sie einen Buchungsbetrag auch mit [↓] anstelle von [RTN] abschließen.

Das Programm AUFGPST

Das Programm AUFGPST ist auf vier Magnetkarten, die dem HP-75 beiliegen, abgespeichert. Nachdem Sie dieses Spielprogramm kennengelernt haben, wird der File AUFGPST das Ziel einer Reihe von Filemanipulationen sein.

Die Größe des Files AUFGPST ist auf den Magnetkarten aufgedruckt – in der deallokatisierten Form beträgt sie etwa 4700 Bytes.* Nach der Initialisierung benötigt das Programm ungefähr 1100 Bytes zusätzlich. Prüfen Sie mit mem [RTN], ob Sie genügend Platz im Speicher zur Verfügung haben.

- Wenn 5800 oder mehr Bytes angezeigt werden, ist genügend Platz im Speicher vorhanden, um das Programm einzulesen und zu starten.
- Wenn eine kleinere Zahl als 5800 angezeigt wird, sollten Sie zusätzlichen Speicherplatz für das Programm schaffen. Löschen Sie dazu einen oder mehrere Files im Speicher.

Führen Sie danach den Befehl COPY aus.

* Die Größe des Programms AUFGPST hängt von der Version ab, die Sie benutzen.

```
>copy card to 'aufgpst'
```

Der Befehl COPY enthält den für das Programm gewünschten Filenamen.

```
Copy from card:  Align & [RTN]
```

Die Antwort: der HP-75 ist zur Ausführung der Kartenleseroperation bereit.

Der HP-75 leitet Sie bei dieser wie bei allen Kartenleseroperationen an, indem er mit der derzeitigen Verzögerungsrate Meldungen anzeigt.

Geben Sie edit 'aufgpst' ein, um den Filepointer auf den File AUFGPST zu stellen. Mit AUFGPST können Sie Ihre Fähigkeit prüfen, sich an sechs Paare verborgener Zeichen (je zwei davon sind identisch), zu erinnern, indem Sie diese in einem 12stelligen Feld identifizieren. Zu Beginn jeder Runde erhalten Sie 100 Punkte. Der HP-75 läßt Sie ein Zeichen nach dem anderen für einen Moment betrachten, und Sie versuchen danach, die Zeichenpaare mit möglichst wenigen Fehlern aufzufinden. Für jede Übereinstimmung erhalten Sie 20 Punkte; jeder Fehler kostet Sie 25 Punkte. Die Runde ist beendet, sobald Sie 0 Punkte erreichen.

Starten Sie das Programm mit [RTN]; der Start des Programms wird durch die Initialisierung leicht verzögert.

```
        AUFGEPASST !
PRGM
```

Der Programmkopf.

```
Spiel für 1 oder 2 Personen:
PRGM
```

Geben Sie 1 [RTN] ein (Einzelspiel).

```
leichtes oder schweres Spiel:
PRGM
```

Wählen Sie den Schwierigkeitsgrad. Wählen Sie zu Beginn den einfachen Schwierigkeitsgrad. Geben Sie 1 [RTN] ein.

```
Ihre Initialen?
PRGM
```

Geben Sie in Groß- oder Kleinbuchstaben Ihre Initialen ein (im Beispiel wird AB benutzt). Mit dem nächsten [RTN] wird das Spiel gestartet. Lesen Sie zuvor noch die nächsten Zeilen.

```
Es geht los, AB. Aufgepasst!
PRGM
```

Diese Meldung wird sofort beim Drücken von [RTN] angezeigt. Der HP-75 wird Ihnen jetzt die zu erinnernden Zeichen für einen Augenblick anzeigen.

```
# X # # # # # # # # # # # Start
PRGM
```

Die Zeichen werden nacheinander «umgedreht».

Nachdem Sie die Zeichen gesehen haben, zeigt der HP-75 die beiden Klammern ([ ]), mit denen Sie Zeichen auswählen werden, Ihre Initialen und Ihre Anfangspunkte:

```
[#]# # # # # # # # # # # AB:100
PRGM
```

Jedes # verbirgt ein Zeichen.

Wählen Sie mit [←] und [→] eine Zeichenposition. Drücken Sie dann die Leertaste, um das versteckte Zeichen umzudrehen.

Beispiel:

```
#[#]# # # # # # # # # AB:100
PRGM
```

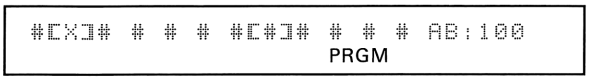
Die Anzeige, nachdem Sie das zweite Zeichen eingestellt haben.

```
#[X]# # # # # # # # # AB:100
PRGM
```

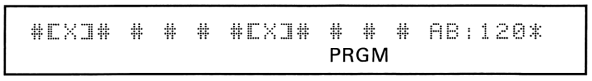
Drücken Sie die Leertaste, um Ihre Wahl anzuzeigen.

Jetzt müssen Sie das Gegenstück zu diesem Zeichen finden. Drücken Sie die Leertaste an der Stelle, wo Sie es vermuten.

Beispiel:



Versuchen wir's hier.



Richtig! Die Punktzahl wird erhöht.

Nach einer Übereinstimmung bleiben die Zeichen angezeigt, Ihre Punktzahl wird erhöht, und Sie erhalten einen Bonus (durch den Stern angezeigt). Bei einem Bonus spielen Sie um doppelte Punktzahl, 40 Punkte anstatt 20. Wenn Sie ein Paar verfehlen, werden die Zeichen wieder verborgen, und Sie verlieren 25 Punkte und das Bonusspiel. Es sind maximal 300 Punkte möglich.

Spielen Sie eine Runde oder zwei, um ein Gefühl für das Spiel zu bekommen. Die schwierige Variante benutzt Interpunktionszeichen anstelle von Buchstaben; und diese werden für eine kürzere Zeit angezeigt. In Anhang F finden Sie ein Listing des Programms AUFGPST; Sie können das Programm variieren, indem Sie die Zeilen 1080 und 1090 ändern. Die Variablen T2, T3, T5 und T6 bestimmen zum Beispiel die Verzögerungszeiten im Programm, N9 setzt die Anzahl der Zahlenpaare, und die Punktbewertung wird mit den Variablen B1, I1, D1 und E1 festgelegt.

Der File AUFGPST wird während der folgenden Filemanipulationen wesentlich geändert.

Verwendung von Filebefehlen bei initialisierten Programmen

BASIC- und Textfiles im Speicher können mit einer Vielzahl von Systembefehlen bearbeitet werden. Ein oder mehrere Files davon können initialisierte (oder allokatisierte) Programmfiles sein – das heißt Programme, die gerade ablaufen oder auf eine Fortsetzung der Ausführung warten. Jeder Filebefehl kann innerhalb eines laufenden Programms oder während einer Unterbrechung des Programms über das Tastenfeld ausgeführt werden. Die folgende Liste beschreibt die Reaktionen in solchen Situationen. Beachten Sie, daß eine Zeile mit Mehrfachanweisungen bei der Ausführung über das Tastenfeld praktisch wie ein Programm reagiert – die Ausführung stoppt vor dem Programmende, wenn einer der Befehle einen Programmhalt bedingen würde (siehe weiter unten).

Befehl	Kommentar
CAT	Der spezifizierte Katalogeintrag wird angezeigt. Nach CAT ALL wird die Ausführung des Programms fortgesetzt, wenn Sie eine andere als eine Pfeiltaste drücken.
CAT CARD	Der Katalogeintrag des Kartenfiles wird angezeigt. Innerhalb eines laufenden Programms wird die Ausführung durch Drücken einer beliebigen Taste mit der nächsten Anweisung fortgesetzt. Nach anderen Kartenleserbefehlen wird die Programmausführung automatisch fortgesetzt, sobald die Kartenleseroperation abgeschlossen ist.
CONT	Das momentane Programm wird bei der spezifizierten Zeile, falls keine Zeile spezifiziert wurde, fortgesetzt.
COPY	Das Kopieren eines initialisierten Programms bedingt dessen Deallokatisierung.
DELETE	Das Löschen einer Programmzeile bedingt eine Deallokatisierung des Programms.
EDIT	Der Katalogeintrag für den EDIT-File wird mit der momentanen Verzögerungsrate angezeigt, und der spezifizierte File kann editiert werden.
FETCH	Die spezifizierte Zeile wird in den Anzeigebuffer eingelesen. Wenn die nächste Anweisung ein INPUT Befehl ist, wird die Zeile angezeigt.
MERGE	Das Mischen eines Programms in ein initialisiertes Programm bedingt immer eine Deallokatisierung des initialisierten Programms.
NAME	Das Benennen eines Programms bedingt dessen Deallokatisierung.
PURGE	Beim Löschen eines Programms wird der gesamte von diesem benutzte Speicher wieder verfügbar.

Befehl	Kommentar
RENAME	Beim Umbenennen eines initialisierten Programms wird dieses deallokatisiert; das Umbenennen eines laufenden Programms hält das Programm an.
RENUMBER	Bewirkt keine Deallokation und keine Unterbrechung eines ablaufenden Programms. In den Anweisungen AUTO, CONT, DELETE, FETCH, LIST, MERGE, PLIST, RENUMBER, PRINT #, READ #, RESTORE # und RUN vorkommende Zeilennummern werden nicht geändert.
RUN	Das Starten eines Programms bedingt eine Deallokation aller initialisierten Programme.
TRANSFORM	Bei der Transformation wird ein initialisiertes Programm deallokatisiert.

Bevor Sie einen der Befehle DELETE, FETCH, MERGE, NAME oder RENUMBER auf einen File anwenden können, müssen Sie zuerst den Filepointer (mit EDIT) auf diesen File stellen.

Das Programm MANAGER

Das Programm MANAGER manipuliert den File AUFGPST mit einer Reihe von Filebefehlen. Prüfen Sie den verfügbaren Speicherplatz mit `mem` RTN; für MANAGER sind 300 Bytes erforderlich. Sie können mit der Deallokation von AUFGPST etwa 1000 Bytes freigeben – geben Sie dazu `clear vars` RTN ein. (Damit werden Programme deallokatisiert und Rechnervariablen gelöscht.)

Führen Sie dann EDIT 'MANAGER' aus , und geben Sie das folgende Programm ein:

```
 10 ! File manager.
 20 DELAY 2 @ WIDTH 32
•30 EDIT 'AUFGPST'


•40 DELETE 1200,1820
 50 DELETE 1850,2420
 60 DELETE 2450,9999
•70 LIST 1830,9999 @ BEEP


•80 INPUT 'Temp. Filename: ';B$

 90 COPY TO B$

 100 CAT B$

 110 RENUMBER 0,10,0,1190

 120 MERGE B$,1110,1190
 130 PURGE B$
 140 EDIT 'MANAGER'

 150 RUN 'AUFGPST'
```

Macht AUFGPST zur Editierung verfügbar. Der Katalogeintrag wird 2 Sekunden lang angezeigt.

Löscht Zeilen aus AUFGPST.

Listet das Ende von AUFGPST. Beschleunigen Sie den Listvorgang mit einer beliebigen Taste.

Geben Sie TEMP als Name eines neuen Files ein.

Kopiert den momentanen File AUFGPST auf TEMP.
Zeigt den Katalogeintrag von TEMP an.

Numeriert den Fileanfang von AUFGPST so um, daß die Zeilennumerierung bei Zeile 0 beginnt.
Mischt einen Teil von TEMP in AUFGPST.
Löscht den temporären File.
Stellt den Filepointer wieder auf das Programm MANAGER und zeigt den Katalogeintrag an.
Startet die neue Version des Programms AUFGPST.

Wenn Sie das Programm ausführen, können Sie die einzelnen Filemanipulationen anhand des obigen Listings verfolgen. Bei jedem Lauf des Programms MANAGER wird der Kopfteil des Programms AUFGPST dupliziert.

Ausschalten des HP-75 (BYE)

Mit dem programmierbaren Befehl BYE wird der HP-75 ausgeschaltet. Beim Wiedereinschalten wird die Programmausführung mit der dem Befehl BYE folgenden Anweisung fortgesetzt.

Beispiel:

```
10 INPUT 'Ihr Passwort: ';A$
20 LOCK A$
30 BYE @ DISP 'Richtig'
```

Das Programm sichert den HP-75 mit einem Paßwort und schaltet ihn aus. Wenn Sie [ATTN] drücken, fragt der HP-75 nach dem Paßwort; wenn es richtig eingegeben wird, wird der Rest von Zeile 30 ausgeführt.

Wenn der HP-75 unter Programmkontrolle ausgeschaltet wurde, werden danach eintreffende Termine ignoriert, bis der HP-75 wieder eingeschaltet ist und die Programmausführung beendet hat.

Wenn der HP-75 bei STANDBY OFF länger als fünf Minuten auf eine Tasteneingabe im Programmablauf warten muß, schaltet er sich selbst aus. Die Anweisung INPUT zum Beispiel verlangt eine Antwort des Tastenfelds, und wenn diese nicht innerhalb von fünf Minuten eintrifft, schaltet sich der HP-75 aus. Die Anweisungen ASSIGN IO und CAT ALL und Kartenleseroperationen erwarten ebenfalls eine Reaktion. Das Programm bleibt bei ausgeschaltetem HP-75 initialisiert und kann mit dem Befehl CONT fortgesetzt werden, nachdem der HP-75 wieder eingeschaltet wurde.

Fehler

Während der Entwicklung und Ausführung eines Programms können vier Fehlerarten auftreten: *Syntaxfehler*, *Initialisierungsfehler*, *Laufzeitfehler* und *Logikfehler*.

Wie Sie gesehen haben, werden Ihre Zeilen bei der Eingabe auf Syntax- (oder Sprach-)fehler geprüft. Sie können Syntaxfehler korrigieren, indem Sie beim Editieren der Zeile Zeichen einfügen, löschen oder ersetzen.

Initialisierungsfehler treten auf, wenn Sie das Schreiben eines Programmes beendet haben und es zu starten versuchen. Solche Fehler sind zum Beispiel fehlende Zeilennummern, doppelt vorhandene benutzerdefinierte Funktionen, ungültige Felddimensionen usw. Sie werden über alle Initialisierungsfehler informiert, bevor ein Programm ausgeführt werden kann.

Die dritte Fehlerart tritt in laufenden Programmen auf. Alle Laufzeitfehler unterbrechen ein Programm und halten es an, außer Sie haben DEFAULT ON gesetzt oder fangen den Fehler mit einer Anweisung ON ERROR (Seite 258) auf. Zu Laufzeitfehlern gehören auch der Zugriff auf ein nicht existierendes Feldelement, der Versuch, nicht initialisierte Daten zu benutzen, Stringüberlauf, Diskrepanzen in READ und DATA Anweisungen, versuchter Zugriff auf einen nicht existenten File usw.

Die vierte Fehlerart ist manchmal am schwierigsten zu lokalisieren. Logische Fehler ergeben sich bei falschem *Entwurf*, nicht bei der falschen *Verschlüsselung* des Programms. Wenn ein Programm bis zum Ende durchläuft, aber falsche oder bedeutungslose Ergebnisse liefert, wenn ein Programmlauf nicht mehr endet, oder wenn einige Eingaben richtige Resultate ergeben und andere nicht, handelt es sich um einen Logikfehler. Die Suche und Beseitigung von Logikfehlern liegt außerhalb der Thematik dieses Handbuchs, aber das Vermeiden logischer Fehler ist grundlegend für gute Programmierung.

In Abschnitt 17, «Fehlersuche», finden Sie das Handwerkszeug zur Behandlung von sowohl Laufzeitfehlern als auch Logikfehlern.

Abschnitt 12

Verzweigungen, Schleifen und Unterprogramme

Inhalt

Einführung

Mit Hilfe von Verzweigungen, Schleifen und Unterprogrammen können Sie die Abarbeitung eines Programmes steuern. In diesem Abschnitt werden folgende Themen behandelt:

- Unbedingte Verzweigung mit GOTO.
- Bedingte Verzweigung mit IF...THEN...ELSE.
- FOR...NEXT Schleifen.
- Erzeugen von Unterprogrammen mit den Anweisungen GOSUB, RETURN und POP.
- Verzweigen mit den Anweisungen ON...GOTO und ON...GOSUB.
- Setzen von Timer-Interrupts mit der Anweisung ON TIMER #.

Verzweigungsanweisungen ändern den Fluß der Programmausführung und können erreichen, daß eine oder mehrere Programmanweisungen übersprungen werden. Eine Programmanweisung, die eine Zeilennummer spezifiziert (zum Beispiel GOTO 100) oder die die Ausführung bei einer anderen als der momentan ausgeführten Zeile fortsetzt (zum Beispiel RETURN) kann nicht über das Tastenfeld ausgeführt werden.

Unbedingte Verzweigungen (GOTO)

Die Anweisung GOTO ist einfach und direkt; die Programmkontrolle wird an die spezifizierte Zeile übertragen.

```
GOTO Zeilennummer
```

Dies wird *unbedingte* Verzweigung genannt, weil bei der Ausführung dieser Anweisung in jedem Fall eine Verzweigung erfolgt.

Beispiel:

```
290 GOTO 3000
```

Überträgt die Ausführung an Zeile 3000.

Mit der Anweisung GOTO können Sie zu Zeilen mit höherer oder niedrigerer Zeilennummer verzweigen und damit Programmschleifen erzeugen.

Bedingte Verzweigungen (IF...THEN...ELSE)

Mit einer *bedingten* Verzweigung versetzen Sie ein Programm in die Lage, eine Entscheidung aufgrund eines Vergleichs oder einer Abfrage zu treffen. Benutzen Sie für bedingte Verzweigungen die Anweisung IF...THEN. Dies ist die einfachste Form der Anweisung:

```
IF numerischer Ausdruck THEN Zeilennummer
```

Wenn die Bedingung wahr ist, das heißt, wenn der *numerische Ausdruck* einen von Null verschiedenen Wert ergibt, wird der THEN-Teil der Anweisung ausgeführt. Wenn die Bedingung falsch ist, das heißt, der numerische Ausdruck besitzt den Wert Null, wird die Ausführung bei der auf die Anweisung IF...THEN folgenden Anweisung fortgesetzt.

```
90 IF A THEN 1200
```

Falls A von Null verschieden ist, wird die Ausführung bei Zeile 1200 fortgesetzt; andernfalls wird die Ausführung bei der nächsten Programmzeile fortgesetzt.

```
90 IF C<.0001 THEN 1200
```

Falls C kleiner als .0001 ist, wird der Vergleichsausdruck als 1 bewertet, und das Programm verzweigt zu Zeile 1200.

```
90 IF A$='Y' OR A$='y' THEN 1200
```

Falls A$='Y' oder 'y' ist, ist der Boolesche Ausdruck wahr, und die Ausführung springt auf Zeile 1200.

Die Anweisung IF...THEN wird am häufigsten mit Vergleichsoperatoren (=, <, >, <=, >=, <> und #) benutzt, auch wenn die Abfrage mit Werten beliebiger arithmetischer oder boolescher Ausdrücke durchgeführt werden kann.

Eine weitere Form der Anweisung IF...THEN ermöglicht die bedingte Ausführung einer Anweisung, ohne notwendigerweise zu verzweigen:

```
IF numerischer Ausdruck THEN [zulässige Anweisung... ] oder [Befehl...]
```

Wenn die Bedingung wahr ist (einen von Null verschiedenen Wert ergibt), werden die spezifizierten Anweisungen ausgeführt. Wenn die Bedingung falsch ist (den Wert Null ergibt), werden die Anweisungen nach THEN übersprungen; die Ausführung wird bei der folgenden Programmzeile fortgesetzt. Zulässige Anweisungen und Befehle können mit dem Symbol @ zu beliebigen Kombinationen, die nur in eine Zeile passen müssen, verbunden und hinter THEN gestellt werden.

Beispiele:

```
440 IF I THEN I=I-1
```

Wenn I nicht den Wert Null hat, wird I dekrementiert; andernfalls wird die Anweisung ignoriert.

```
5 IF N=2 THEN ASSIGN IO ':tv,:p'
```

Wenn N gleich 2 ist, wird die HP-IL Schleife der Anweisung entsprechend zugewiesen.

```
1010 IF MEM<C THEN PURGE C$
```

Wenn weniger Speicherplatz verfügbar ist, als mit C spezifiziert, wird der angegebene File gelöscht.

Alle Befehle und die meisten BASIC-Anweisungen können als bedingte Anweisungen spezifiziert werden. Die folgenden Anweisungen sind nach THEN nicht zulässig:

FOR	IF
NEXT	IMAGE
DATA	INTEGER
DEF FN	OPTION BASE
DIM	REAL
END DEF	SHORT
ON TIMER	ON ERROR

Wenn Sie eine der obenstehenden Anweisungen mit IF...THEN verwenden, erhalten Sie Fehler 86 – illegal context. Beachten Sie, daß Sie eine Anweisung IF...THEN, die keine Verzweigung enthält, über das Tastenfeld ausführen können.

Beispiel:

```
>if 1>0 then disp 'wahr'
```

```
wahr
```

Mit RTN wird die IF - Bedingung ausgewertet und die Anweisung THEN ausgeführt.

Die Option ELSE

Die Anweisung IF...THEN kann noch ein drittes Schlüsselwort enthalten: ELSE. Wenn der numerische Ausdruck falsch bewertet wird, führt das Programm die nach ELSE angegebene Anweisung durch. Sie haben bei der Anweisung IF...THEN...ELSE eine Vielzahl von Spezifikationsmöglichkeiten:

```
IF numerischer Ausdruck THEN Zeilennummer                ELSE Zeilennummer
                             [zulässige Anweisung...      [zulässige Anweisung...
                                    oder                         oder
                              Befehl...]                   Befehl...]
```

Beispiele:

```
490 IF I=1 THEN E$='$' ELSE E$=
'%'
```

E nimmt abhängig vom Wert von I einen von zwei Werten an.

```
4560 IF ABS(X)<.001 THEN RETURN
```

Abhängig vom Wert von X wird entweder das Unterprogramm beendet oder eine Verzweigung zu Zeile 4400 ausgeführt.

```
620 IF A*B#0 THEN DEFAULT OFF EL
SE DEFAULT ON
```

Die Werte von A und B bestimmen, wie Fehler behandelt werden.

Die Beschränkungen für THEN gelten auch für ELSE – ELSE kann mit beliebigen Systembefehlen und zulässigen BASIC-Anweisungen oder Kombinationen dieser beiden oder mit einer Zeilennummer ergänzt werden.

Mehrfachanweisungen nach THEN und ELSE (@)

Sie können mit den Anweisungen THEN und ELSE mehr als eine Anweisung ausführen, wenn Sie die Befehle und Anweisungen mit dem Zeichen @ verknüpfen.

Beispiele:

```
120 IF A<1 THEN BEEP @ DISP 'Za
hl zu klein.' @ goto 110
```

Falls die Bedingung erfüllt ist, werden alle drei Anweisungen ausgeführt.

```
2000 IF T=-1 THEN STOP ELSE A=0
@ BEEP 700
```

Falls T ungleich –1 ist, wird ein Tonsignal erzeugt und A=0 gesetzt.

Zusammenfassung: Wenn die Bedingung erfüllt ist, werden alle Anweisungen nach THEN ausgeführt. Wenn die Bedingung nicht erfüllt ist, werden alle Anweisungen nach ELSE ausgeführt. Die Länge einer Anweisung IF...THEN...ELSE mit Mehrfachanweisungen ist nur durch die Zeilenlänge des HP-75 begrenzt. Beachten Sie, daß die Anweisung IF...THEN keine weitere Anweisung IF...THEN enthalten darf.

Schleifen (FOR...NEXT)

Einen Anweisungsblock, der mehrfach hintereinander ausgeführt wird, bezeichnet man als *Schleife*. Die Schleife im Programm KONTO bewirkt, daß die Ausführung fortgesetzt wird, bis sie mit ATTN unterbrochen wird oder bis die Bedingung nach IF falsch wird, und daher die Schleife beendet wird.

Mit der Anweisung FOR-NEXT besitzen Sie ein einfaches und wirkungsvolles Mittel zur Schleifensteuerung. Die Anweisungen FOR und NEXT stehen an Anfang und Ende einer Reihe von Anweisungen, die dann eine vorgegebene Anzahl von Malen ausgeführt werden.

```
FOR Schleifenzähler = Anfangswert TO Endwert
    ⋮
STEP Inkrementwert
```

Die Anweisung FOR bestimmt den Anfang der Schleife und die Anzahl der Schleifendurchläufe. Der *Schleifenzähler* muß eine einfache numerische Variable sein, wie zum Beispiel I oder I1. Die Parameter *Anfangswert*, *Endwert* und *Inkrementwert* können beliebige numerische Ausdrücke sein. Wenn kein STEP *Inkrementwert* spezifiziert wird, wird der Schleifenzähler bei jedem Schleifendurchgang um 1 erhöht.

Die Anweisung FOR hat folgende Wirkung:

- Setzen des Schleifenzählers auf den Anfangswert.
- Abspeichern des Endwerts für die Schleifenzählung.
- Abfrage der Bedingung für das Verlassen der Schleife durch Vergleich des Schleifenzählers mit dem Endwert.

Solange der Wert des Schleifenzählers kleiner als der Endwert ist (bei positivem Inkrementwert), wird die Programmausführung mit der nächsten Anweisung nach FOR fortgesetzt.

Die Anweisung NEXT hat folgende Wirkung:

- Inkrementierung (oder Dekrementierung) des Schleifenzählers.
- Rückgabe der Programmausführung an die Abfrage in der FOR-Anweisung und dadurch Definition des Schleifenendes.

Die Schleife wird verlassen und die Programmkontrolle an die auf NEXT folgende Anweisung übergeben, sobald der Wert des Schleifenzählers größer als der Endwert (bzw. kleiner bei negativem Inkrement) der Anweisung FOR wird.

Die Anfangs- und Endwerte in der FOR Anweisung werden nur einmal berechnet – beim ersten Eintritt des Programms in die Schleife. Der Endwert der Schleife kann nicht innerhalb der Schleife geändert werden. Wenn der Anfangswert des Schleifenzählers größer (oder bei negativem Inkrement kleiner) als der Endwert ist, wird nur die FOR Anweisung in der Schleife ausgeführt.

```
 10 FOR I=1 TO 28
• 20 DISP STR$(MOD(I,10));


 30 NEXT I
• 40 DISP I
```

Die Funktion STR$ zeigt den Wert von (MOD(I,10)) ohne führende und nachlaufende Leerstellen an (Seite 201).

Zeigt den Wert von I nach dem Verlassen der Schleife an.

Setzen Sie DELAY auf 0 Sekunden und WIDTH auf 32 oder mehr Spalten, bevor Sie das Programm starten. Das Programm erzeugt folgende Anzeige:

```
12345678901234567890123456789 29
```

Das Ergebnis zeigt, daß die Anweisung NEXT den Schleifenzähler über den Endwert hinaus erhöht.

Hinweis: Dieses Programm kann nach einer einfachen Änderung die 32 Spaltenpositionen des Anzeigefensters des HP-75 aufzeigen. Ändern Sie Zeile 10 in FOR I=1 TO 32, und ändern Sie Zeile 40 in DISP.

Sie können eine FOR...NEXT Schleife über das Tastenfeld durchführen, wenn die gesamte Schleife mit @-Zeichen verbunden wird und wenn nicht aus der Schleife «heraus» verzweigt wird.

Beispiel:

```
>for i=1 to 5@beep i*200@next i
```

Durch Drücken von RTN wird die Anweisung BEEP fünfmal hintereinander ausgeführt.

Sie können FOR...NEXT Schleifen daher in Tastendefinitionen verwenden.

Ändern des Inkrementwertes (STEP)

Die Schleifenzählervariable einer FOR . . . NEXT Schleife kann beliebige Zahlenwerte einschließlich negativer und dezimaler Werte annehmen. Mit dem Schlüsselwort STEP in der Anweisung FOR können Sie einen beliebigen Inkrement- oder Dekrementwert für den Schleifenzähler vorgeben.

Beispiel: Zeigen Sie die Dezimalbrüche von 0.5 bis 0, in Schritten von 1/16 oder 0.0625 abnehmend, an.

```
• 10 FOR I=.5 TO 0 STEP -.0625
  20 DISP I;
  30 NEXT I
```

Spezifiziert eine negative Dezimalzahl als STEP-Wert.

Der Anfangswert von I ist 0.5. Bei jeder Ausführung von NEXT I wird I um 0.0625 vermindert. Sobald I kleiner als der Endwert (0) wird, wird die Schleife verlassen. Das Programm erzeugt zwei Ausgabezeilen (bei WIDTH 32):

```
 .5   .4375   .375   .3125   .25
 .1875   .125   .0625   0
```

Die Schleife wird neunmal durchlaufen.

Hinweis: Am Programmende wird der Cursor nicht mit einem Wagenrücklauf an den Zeilenanfang gesetzt, da die Anweisung DISP in Zeile 20 mit einem Semikolon endet. Deshalb erscheint das nächste über das Tastenfeld eingegebene Zeichen hinter den derzeitigen Anzeigedaten in der gleichen Zeile. Löschen Sie die Zeile mit ATTN oder EDIT.

Wenn der Endwert größer oder gleich dem Anfangswert und das Inkrement 0 ist, erzeugt die FOR Anweisung eine Endlosschleife.

Verschachtelte Schleifen

Wenn eine Schleife vollständig in einer anderen enthalten ist, wird die innere Schleife als *verschachtelt* bezeichnet. Eine Schleife kann innerhalb einer verschachtelten Schleife enthalten sein (bis zu 255 Ebenen), solange die Schleifen sich nicht überschneiden.

Beispiel:

Unzulässige Verschachtelung

```
10 DISP 'I','J'
20 FOR I=1 TO 3
30 FOR J=1001 TO 1003
40 DISP I,J
50 NEXT I
60 NEXT J
```

Zulässige Verschachtelung

```
10 DISP 'I','J'
20 FOR I=1 TO 3
30 FOR J=1001 TO 1003
40 DISP I,J
50 NEXT J
60 NEXT I
```

Die falsch verschachtelte Schleife erzeugt Fehler 47 – no matching FOR – in Zeile 60, wenn die I Schleife verlassen wird. Das obenstehende, richtig verschachtelte Beispiel erzeugt folgende Anzeige:

```
I                    J
 1                    1001
 1                    1002
 1                    1003
 2                    1001
 2                    1002
 2                    1003
 3                    1001
 3                    1002
 3                    1003
```

Programmerklärung. Die Schleife J wird beendet, bevor I inkrementiert wird. Genauer gesagt, sobald J in der Anweisung NEXT J auf 1004 erhöht wird, wird die Programmausführung bei NEXT I fortgesetzt. Bei jeder Ausführung von FOR J... wird J auf 1001 initialisiert. Sobald I den Wert 4 erreicht, wird die äußere Schleife verlassen und das Programm beendet.

Sie können aus einer FOR-NEXT Schleife heraus verzweigen, aber Verzweigen in eine Schleife hinein erzeugt Fehler 47 – no matching FOR, wenn die NEXT Anweisung ausgeführt wird, bevor das Programm die dazugehörige FOR Anweisung ausgeführt hat.

Unterprogramme (GOSUB, RETURN)

Oft muß die gleiche Anweisungsfolge mehrmals an verschiedenen Stellen eines Programms ausgeführt werden. Mit Hilfe eines Unterprogramms brauchen Sie eine Gruppe von Anweisungen nur einmal einzugeben, und können dann von verschiedenen Stellen innerhalb des Programms auf diese Anweisungen zugreifen. Wenn Sie alle oft benutzten Routinen ans Ende Ihres Programmes stellen, wird die Struktur des Programmes leichter verständlich.

```
GOSUB Zeilennummer
```

Die Anweisung GOSUB übergibt die Programmausführung an das gewünschte Unterprogramm. Die *Zeilennummer* ist die Nummer der ersten Zeile des Unterprogramms.

Beispiel:

```
90 GOSUB 800
```

Nach dieser Anweisung wird die Programmausführung bei Zeile 800 fortgesetzt.

Es ist ein gängiger Programmiergebrauch, in der Zeile vor der ersten Zeile eines Unterprogramms einen Kommentar einzufügen, um die Aufgabe des Unterprogramms anzudeuten.

Die zuletzt ausgeführte Anweisung eines Unterprogramms muß die Anweisung RETURN sein.

```
RETURN
```

Sobald ein RETURN gefunden wird, wird die Programmkontrolle an die nächste Anweisung nach der zugehörigen GOSUB Anweisung weitergegeben. Diese Anweisung kann in der gleichen Zeile wie die Anweisung GOSUB oder in der Folgezeile stehen.

Beispiel:

```
90 GOSUB 800 @ DISP 'Zuhause'
```

Wenn das Unterprogramm nach Zeile 800 das zugehörige RETURN antrifft, wird die Ausführung mit der Anweisung DISP fortgesetzt.

Alle Variablen des Hauptprogramms sind in Unterprogrammen verfügbar. Wird der Wert einer Variablen in einem Unterprogramm geändert, bleibt er auch im Hauptprogramm geändert.

Unterprogramme können *verschachtelt* werden; das heißt, die Programmausführung kann in ein zweites Unterprogramm verzweigen, bevor die Anweisung RETURN des ersten ausgeführt wurde. Es sind bis zu 255 Verschachtelungsebenen möglich. Wenn ein RETURN angetroffen wird, kehrt die Programmkontrolle in die Zeile der letzten GOSUB Anweisung zurück und führt die unmittelbar auf GOSUB folgende Anweisung aus.

Die berechnete GOTO Anweisung (ON...GOTO)

Mit der berechneten GOTO Anweisung können Sie die Programmkontrolle an eine von mehreren Programmzeilen, die vom Wert des numerischen Ausdrucks abhängt, übertragen.

```
ON numerischer Ausdruck GOTO Zeilennummer [, Zeilennummer...]
```

Der numerische Ausdruck wird berechnet und auf einen ganzzahligen Wert gerundet. Ist der Wert 1, wird die Ausführung bei der ersten spezifizierten Zeile fortgesetzt; beim Wert 2 wird zur zweiten spezifizierten Zeile verzweigt usw.

Beispiel:

```
200 ON R GOTO 25, 80, 150
```

Diese Anweisung besagt: springe nach Zeile 25, wenn R=1; springe nach Zeile 80, wenn R=2; und springe nach Zeile 150, wenn R=3. Besitzt R einen gerundeten Wert kleiner als 1 oder größer als die Anzahl der Zeilen in der Liste, erhalten Sie bei der Ausführung der Anweisung Fehler 11 – arg out of range.

Die berechnete GOSUB Anweisung (ON...GOSUB)

Mit einer berechneten Anweisung GOSUB können Sie über einen numerischen Ausdruck ein Unterprogramm auswählen, das die Programmausführung fortsetzen soll. ON...GOSUB überträgt die Programmkontrolle an die erste Anweisung eines Unterprogramms und wirkt sonst genau wie die Anweisung ON...GOTO.

ON *numerischer Ausdruck* GOSUB *Zeilennummer* [, *Zeilennummer...*]

Der numerische Ausdruck wird berechnet und auf einen ganzzahligen Wert gerundet. Beim Wert 1 wird auf die erste Zeilennummer in der Liste zugegriffen usw.

Die Anweisung RETURN des Unterprogramms gibt nach der Ausführung die Programmkontrolle an die der Anweisung ON...GOSUB unmittelbar folgende Anweisung weiter.

Beispiel:

```
  100 INPUT 'Argument, 1-3: ';A,I

• 200 ON I GOSUB 500, 700, 900

  300 BEEP
  400 STOP
• 500 DISP SIN(A)
  600 RETURN
• 700 DISP COS(A)
  800 RETURN
• 900 DISP TAN(A)
  1000 RETURN
```

Diese Anweisung besagt:
Falls I=1 GOSUB 500.
Falls I=2 GOSUB 700.
Falls I=3 GOSUB 900.

Erstes Unterprogramm.

Zweites Unterprogramm.

Drittes Unterprogramm.

Die Anweisung RETURN in jedem der obenstehenden Unterprogramme gibt die Programmkontrolle an die erste Anweisung nach ON...GOSUB zurück, – in diesem Fall an die Anweisung BEEP in Zeile 300.

Wenn der numerischen Ausdruck auf einen Wert kleiner als 1 oder größer als die Anzahl von Zeilen in der Liste gerundet wird, erhalten Sie Fehler 11 – arg out of range.

Umgehen einer anstehenden Rücksprungbedingung (POP)

Bei der Ausführung von GOSUB wird eine anstehende Rücksprungbedingung erzeugt, die die Stelle «markiert», wohin die Programmausführung bei der nächsten RETURN Anweisung zurückkehren soll. Wenn mehrmals hintereinander ein GOSUB ohne RETURN ausgeführt wird, erhalten Sie eine entsprechende Anzahl von Unterprogrammebenen und anstehenden Rücksprüngen.

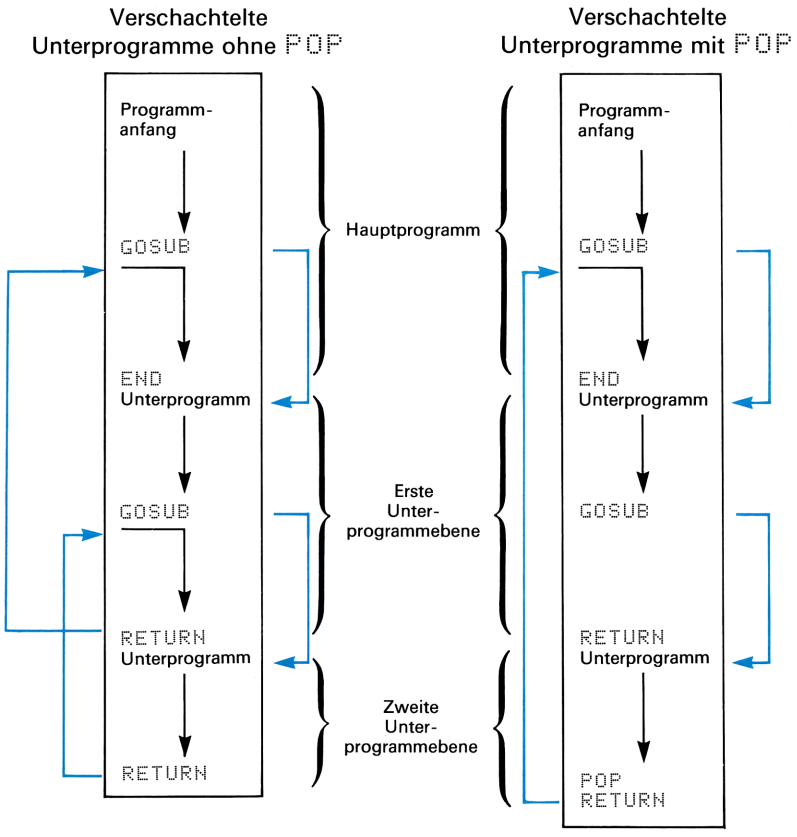
Manchmal ist es jedoch wünschenswert, den Programmablauf so zu ändern, daß ein oder mehrere Unterprogrammrücksprünge ausgelassen werden. Wenn Sie dazu einfach um RETURN-Anweisungen herum verzweigen, entstehen folgende Fehlermöglichkeiten:

- Nachfolgende RETURN-Anweisungen bewirken, daß das Programm an einer falschen Stelle fortgesetzt wird.
- Sie erhalten tiefer als auf 255 Ebenen verschachtelte Unterprogramme.

Solche Fehler können Sie mit der Anweisung POP verhindern, die einfach den aus der letzten GOSUB Anweisung anstehenden Rücksprung aufhebt.

```
POP
```

Mit jeder POP Anweisung wird eine Unterprogrammebene umgangen.



Beispiel: Das folgende Programm kann folgendermaßen unterteilt werden:

- Hauptprogramm – Zeilen 10 bis 40.
- Erstes Unterprogramm – Zeilen 50 bis 80.
- Zweites Unterprogramm – Zeilen 90 bis 120.

Wenn das Programm ohne die Anweisung POP in Zeile 110 ausgeführt wird, folgt der Programmablauf dem Diagramm links in der vorhergehenden Abbildung; wenn die Anweisung POP ausgeführt wird, folgt der Programmablauf dem Diagramm rechts.

```
 10 DISP 'Hauptprogramm'
 20 GOSUB 50 ! zum ersten
 Unterprogramm
 30 DISP 'Zurueck im Hauptprogram
 m' ! vom 1. oder 2. Unterprogram
 m.
 40 STOP
•50 DISP ' 1. Unterprogramm'
 60 GOSUB 90 ! zum 2. Unterprogra
 mm.
 70 DISP ' Zurueck im 1. Unterpr
 ogramm' ! vom 2. Unterprogramm.
 80 RETURN ! zum Hauptprogramm.
•90 DISP '  2. Unterprogramm'
•100 INPUT '   pop? j/n: ';A$

•110 IF A$='j' THEN POP

 120 RETURN ! bei (J) zum 1. Unte
 rprogramm; bei (N) zum Hauptprog
 ramm.
```

Geben Sie den String mit zwei führenden Leerstellen ein.

Geben Sie den String mit vier führenden Leerstellen ein.

Wenn der Benutzer die Option POP wählt, wird der anstehende Return umgangen.

Setzen Sie DELAY 1 und starten Sie das Programm:

```
Hauptprogramm
  1. Unterprogramm
    2. Unterprogramm
    pop? j/n: n
  Zurueck im 1. Unterprogramm
Zurueck im Hauptprogramm
```

Wählen Sie die Version ohne POP.

Starten Sie das Programm noch einmal, dieses Mal mit der Anweisung POP:

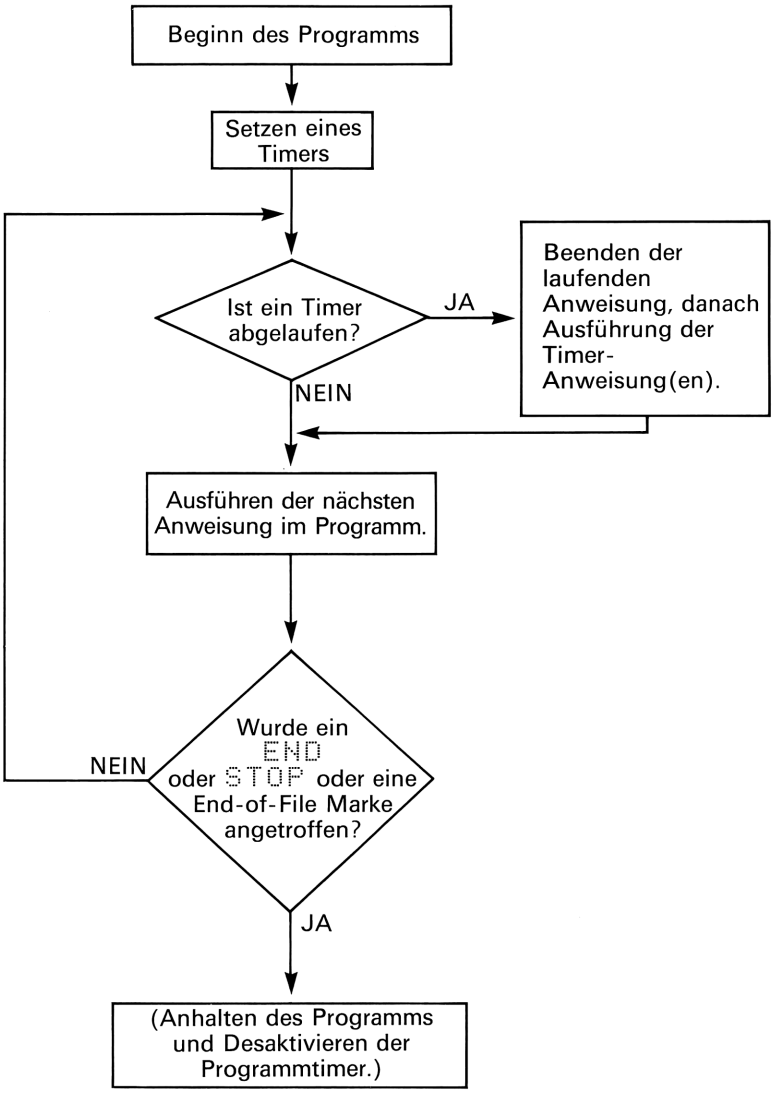
```
Hauptprogramm
  1. Unterprogramm
    2. Unterprogramm
    pop? j/n: j
Zurueck im Hauptprogramm
```

POP annulliert den letzten anstehenden Rücksprung.

Die POP-Anweisung kann wiederholt ausgeführt werden, um ein viele Ebenen tief verschachteltes Unterprogramm zu verlassen. Wenn die Anzahl der ausgeführten POP-Anweisungen die Anzahl der Unterprogrammebenen überschreitet, und danach ein RETURN auftritt, erhalten Sie Fehler 50 – RETURN w/o GOSUB. Die Anweisung RETURN erzeugt dann einen Fehler, da keine anstehende Rücksprungbedingung mehr gegeben ist.

Programmtimer

Sie können in einem Programm des HP-75 einen oder mehrere Timer (Zeitgeber) setzen, um einen beliebigen Befehl oder eine zulässige Anweisung nach bestimmten Zeitintervallen auszuführen. Das folgende Flußdiagramm zeigt, wie ein Timer den Programmablauf steuert.



Setzen eines Timers (ON TIMER #)

Benutzen Sie die Anweisung ON TIMER #, wenn Sie einen der 1001 verfügbaren Programmtimer setzen wollen.

ON TIMER # *Timernummer*, *Sekunden* [*zulässige Anweisung...*] oder [*Befehl...*]

Die Timernummer. Die Timernummer kann ein beliebiger numerischer Ausdruck sein, dessen gerundeter Wert zwischen 0 und 1000 einschließlich liegt. Wenn eine existierende Timernummer für eine weitere Anweisung ON TIMER später im Programm nochmals verwendet wird, ersetzt der zweite Timerwert den ersten.

Sekunden. Die Anzahl Sekunden kann durch einen beliebigen numerischen Ausdruck spezifiziert werden. Timerunterbrechungen sind auf die Zehntelsekunde genau. Die Minimalzeit, die Sie spezifizieren können, beträgt 0.1 Sekunden; kürzere Zeitparameter und negative Werte werden durch 0.1 Sekunden ersetzt.

Die Maximalzeit, die Sie spezifizieren können, ist praktisch unbegrenzt und liegt im Bereich von Jahrhunderten (2^{41}–1 Sekunden).

Zulässige Anweisung oder Befehl. Obwohl viele Timeranwendungen Verzweigungen mit GOTO oder GOSUB beinhalten, kann jede BASIC-Anweisung, die nach dem Schlüsselwort THEN erlaubt ist (Seite 178) und jeder Systembefehl in der Anweisung ON TIMER # spezifiziert werden. Es können auch mehrere Instruktionen folgen, wenn sie mit dem Zeichen @ verknüpft werden.

Beispiel:

```
10 ON TIMER # 1, 1 DISP TIME$

20 GOTO 20
```

Setzt den Timer #1, so daß er das Programm jede Sekunde unterbricht und die Zeit anzeigt. Leerschleife zum Warten auf den Timer-Interrupt.

Ein *Timer-Interrupt* (Unterbrechung durch einen Timer) wird verzögert, bis alle Anweisungen der momentanen Zeile ausgeführt worden sind (ein Timer-Interrupt unterbricht aber eine Zeile mit Mehrfachanweisungen, wenn diese die Anweisungen GOSUB, NEXT, CALL, GOTO oder CONT enthält). Nachdem der Timer seine eigenen Instruktionen ausgeführt hat, wird das Programm fortgesetzt, als ob keine Unterbrechung stattgefunden hätte. Die Ausführung kehrt also zu der Zeile zurück, die auf die vor der Unterbrechung zuletzt ausgeführte Zeile folgt. Im obenstehenden Beispiel wird die Programmausführung also bei Zeile 20 fortgesetzt, weil dies die normalerweise nach der Anweisung GOTO ausgeführte Anweisung ist.

Die Anweisung END oder eine End-of-File Marke löscht die im Programm gesetzten Timer. Wenn Sie die Programmausführung mit ATTN oder STOP anhalten, werden die Timer dieses Programms desaktiviert. Mit CONT werden die Timer mit ihrem letzten Zählerinhalt wieder aktiviert.

Mit Programmtimern haben Sie die Möglichkeit, den HP-75 in bestimmten Zeitintervallen *einzuschalten*, wenn er zuvor unter Programmkontrolle ausgeschaltet worden ist. Wenn ein Programm den Befehl BYE ausführt, wird der HP-75 ausgeschaltet, aber alle derzeitigen Timer bleiben aktiv. Beim nächsten Timersignal schaltet sich der HP-75 wieder ein, führt die Instruktionen des Timers aus und setzt die Programmausführung mit der dem Befehl BYE folgenden Anweisung fort.

Beispiel:

```
10 ON TIMER # 1,2 BEEP

20 BYE
30 DISP TAB(12);'An!'
```

Timer 1 unterbricht das Programm alle 2 Sekunden; dies tritt bei ausgeschaltetem HP-75 ein. Die Anweisung BEEP wird ausgeführt; das Programm bei Zeile 30 fortgesetzt.

Außer in dieser Situation – wo ein Timer bei ausgeschaltetem HP-75 aktiv bleibt – kann ein Timer das Programm, von dem er gesetzt wurde, nur unterbrechen, solange das Programm aktiv ist. Wenn ein Programm ein anderes aufruft, zählen die Timer des ersten Programms weiter, aber sie erzeugen keine Unterbrechung, solange das zweite Programm läuft. Siehe auch Seite 230, «Aufrufen von Programmen».

Ausschalten eines Timers (OFF TIMER #)

Mit der Anweisung OFF TIMER # kann ein Programm einzelne Timer ausschalten.

```
OFF TIMER # Timernummer
```

Ein Programm desaktiviert einen Timer, wenn eine OFF TIMER # Anweisung für den betreffenden Timer ausgeführt wird. Solange dieser Timer nicht wieder mit einer weiteren Anweisung ON TIMER # gesetzt wird, kann er keine weiteren Unterbrechungen mehr erzeugen.

Timer-Interrupts und Verzweigungen (ON TIMER #...GOTO und ON TIMER #...GOSUB)

Programmtimer werden häufig dazu benutzt, in bestimmten Zeitabständen ganze Routinen auszuführen. Dazu dienen die Anweisungen ON TIMER #...GOTO und ON TIMER #...GOSUB.

GOTO. Die Anweisung GOTO in einer ON TIMER #...GOSUB Anweisung überträgt die Programmkontrolle an eine andere Zeile im Programm. Sobald der spezifizierte Timer einen Interrupt erzeugt, verzweigt die Programmausführung vom Ende der momentan ausgeführten Zeile zum Anfang der spezifizierten Zeile.

Beispiel:

```
• 10 ON TIMER # 5, 10 GOTO 500
  20 GOTO 20
  500 DISP 'Zehn Sekunden'
  510 BEEP 100
• 520 DISP 'Fortsetzung'
```

Beim Interrupt wird zu Zeile 500 verzweigt.

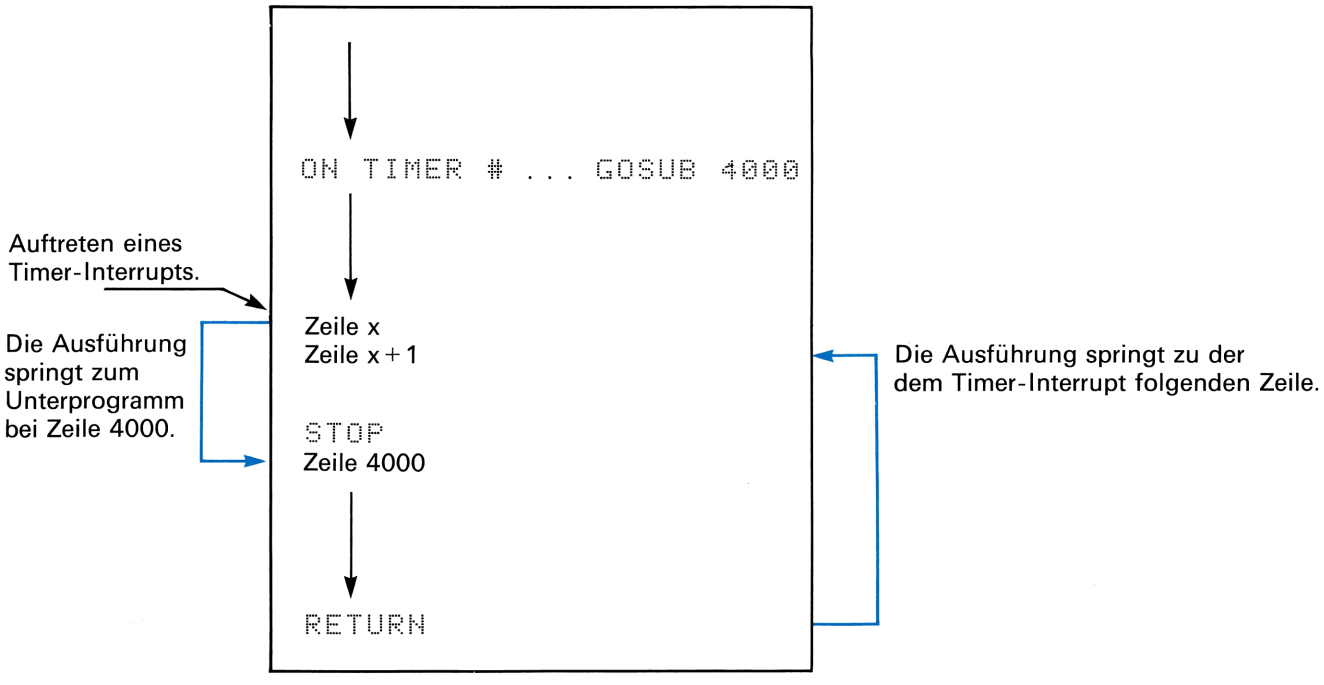
Die Ausführung wird wie nach jeder anderen unbedingten Verzweigung fortgesetzt.

Sie sollten bei der ON TIMER # Anweisung keine weiteren Anweisungen nach GOTO anhängen, da diese niemals ausgeführt werden.

```
ON TIMER # 5, 30 GOTO 500 @ DISP 'Wo?'
                ausgeführt   ignoriert
```

Nach der Anweisung ON TIMER #...GOTO kann die Ausführung des Programms nicht mehr dorthin zurückkehren, wo sie unterbrochen wurde.

GOSUB. Wenn Sie in der Anweisung ON TIMER # als zweites Schlüsselwort GOSUB verwenden, kehrt die Programmausführung wieder dorthin zurück, wo die Unterbrechung wirksam wurde – das heißt zu der auf die Unterbrechung folgenden Zeile – sobald das Timer-Unterprogramm abgearbeitet ist.



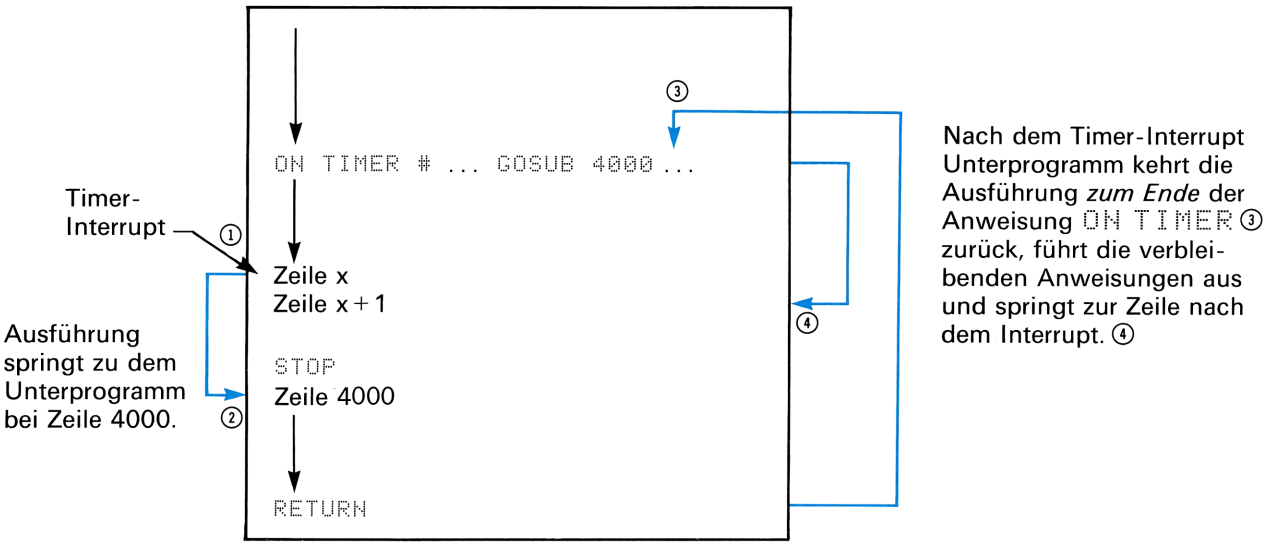
Wenn die Programmausführung nach der Anweisung `ON TIMER #` auf ein `RETURN` stößt, wird sie mit der Zeile nach der Unterbrechung fortgesetzt.

Sehen Sie, was passiert, wenn Sie bei der Anweisung `ON TIMER #` noch weitere Aweisungen an `GOSUB` anschließen.

```
ON TIMER # 5, 30 GOSUB 4000 @ DISP 'hier'
```

Zuerst ausgeführt — nach dem Unterprogramm ausgeführt

Dazu eine Illustration:



Das heißt, wenn nach der Anweisung `ON TIMER #...GOSUB` ein `RETURN` ausgeführt wird, kehrt die Ausführung zur Anweisung `ON TIMER #` zurück, führt die in dieser Zeile stehenden weiteren Instruktionen aus, und verzweigt *dann* zur Zeile nach der Unterbrechung.

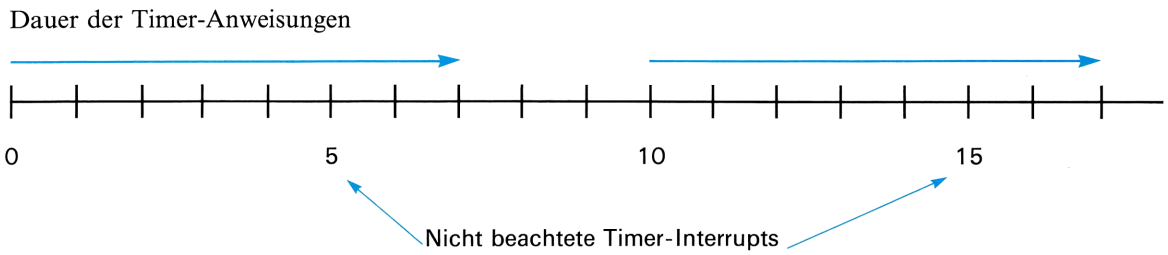
Mehr über Timer

Timer-Interrupt Routinen können wie Unterprogramme verschachtelt werden. Wenn dann innerhalb einer Timer-Interrupt Routine eine andere Timer-Unterbrechung fällig wird, wird der zweite Timer-Interrupt ausgeführt und der noch nicht ausgeführte Teil der ersten Routine wird als anstehendes Unterprogramm behandelt. Nachdem die Routine des zweiten Timers beendet ist, wird die erste Timerroutine fortgesetzt.

Wenn in einem Programm versucht wird, Timer-Interrupts mehr als 255 Ebenen tief zu verschachteln, tritt Fehler 49 – `GOSUB overflow` – auf. Wenn die Timer-Interrupts so schnell hintereinander auftreten, daß das System nichts anderes mehr bearbeiten kann, kann eine nicht beabsichtigte Timer-Verschachtelung entstehen, die irgendwann einmal das Maximum von 255 Ebenen überschreitet und den Fehler `GOSUB overflow` erzeugt.

Timer-Routinen dürfen keine `ON TIMER #` Deklarationen enthalten. Ein Versuch erzeugt Fehler 86 - `illegal context`.

Ein Timer kann sich nicht selbst unterbrechen. Wenn ein Timer zum Beispiel alle 5 Sekunden unterbricht, die Ausführung der `ON TIMER`-Anweisungen aber 7 Sekunden benötigt, werden die nachfolgenden Unterbrechungen dieses Timers ignoriert, bis dessen Instruktionen abgearbeitet worden sind.



Obwohl der Timer alle 5 Sekunden abläuft, wird die Timer-Routine nur alle 10 Sekunden ausgeführt.

Jeder Timer-Interrupt wird in Wirklichkeit wie ein Unterprogramm ausgeführt; das heißt, jede ON TIMER # Anweisung erzeugt eine anstehende Unterprogrammbedingung, die nach der Ausführung aller Timerinstruktionen abgearbeitet wird. Mit Hilfe der anstehenden Unterprogrammbedingung kann das Programm nach der Ausführung der Timer-Routine bei der Zeile nach der Unterbrechungsstelle fortgesetzt werden. Die Anweisung RETURN ist dabei nicht notwendig, da jede ON TIMER Anweisung eine unsichtbare RETURN Anweisung beinhaltet, die nach der letzten Anweisung des Timers ausgeführt wird. Sie können in der Anweisung ON TIMER # ein RETURN explizit anführen; dies hat dann die gleiche Wirkung wie das unsichtbare RETURN.

Beispiel:

```
100 ON TIMER # 1,30 BEEP@RETURN
```

Die Anweisung RETURN beendet die Timeranweisung. Sie ist jedoch nicht erforderlich.

Eine ähnliche anstehende Unterprogrammbedingung wird erzeugt, wenn die Programmausführung an eine Anweisung ON ERROR (Abschnitt 17, Seite 260) übertragen wird.

Wenn ein Programmtimer während der Anweisung INPUT abläuft (bevor Sie RTN gedrückt haben), werden die Timer-Instruktionen *nach* der Zuweisung der Tastenfeld-Eingaben auf die INPUT Variablen ausgeführt.

Abschnitt 13

Felder, Strings und benutzerdefinierte Funktionen

Inhalt

Einführung

Eine Feldvariable (oder einfach ein Feld) ist eine Sammlung von gleichartigen Datenelementen unter einem gemeinsamen Namen. Beim HP-75 können Felder ein- oder zweidimensional sein und Zahlenwerte der Genauigkeit REAL, SHORT und INTEGER aufnehmen.

Ein eindimensionales Feld kann als eine Liste oder Spalte von Daten betrachtet werden. Ein zweidimensionales Feld (oft auch Matrix genannt) ist eine Wertetabelle, die mehrere Zeilen und Spalten besitzen kann.

Spalte	**1**	**2**	**3**	...	*c*
Zeile 1	a_{11}	a_{12}	a_{13}	...	a_{1s}
2	a_{21}	a_{22}	a_{23}	...	a_{2s}
3	a_{31}	a_{32}	a_{33}	...	a_{3s}
⋮	⋮	⋮	⋮		⋮
z	a_{z1}	a_{z2}	a_{z3}	...	a_{zs}

Ein zweidimensionales Feld besitzt *zxs* Elemente, wobei *z* die Anzahl Zeilen und *s* die Anzahl Spalten angibt.

Die Indizes jedes Elements a_{ij} kennzeichnen die Zeile (*i*) und Spalte (*j*) seiner Position.

Die Liste der Zahlen 117.50, 403.50, 514.79 und 603.48 (angeordnet von 1 bis 4) kann als Gruppe gleichartiger Datenelemente, nämlich reeller Zahlen, aufgefaßt werden. Diese Liste wird sinnvollerweise in einem eindimensionalen Feld angelegt. Wir behandeln die Liste als Paare von Werten und *Indices*. Indices geben die Position der Werte oder *Elemente* im Feld an. Wenn die Feldvariable S( ) genannt wird, können die einzelnen Elemente von S( ) über Indices spezifiziert werden, wie zum Beispiel S(2) oder S(3).

Die Werte der folgenden Tabelle können in einem zweidimensionalen Feld abgespeichert werden:

Tag	Tiefst-temperatur	Höchst-temperatur	Niederschlags-menge
21	12	20	.51
22	10	17	.02
23	–6	12	.00

Diese Tabelle besteht aus 3 Zeilen und 4 Spalten und enthält damit zwölf Werte. Die Elemente der Tabelle müssen mit Zeilen- *und* Spaltenindices (*z,s*) spezifiziert werden. Wenn die Indexwerte mit 1 beginnen, können die Elemente so identifiziert werden:

```
D(1,1)=21   D(1,2)=12   D(1,3)=20   D(1,4)=.51
D(2,1)=22   D(2,2)=10   D(2,3)=17   D(2,4)=.02
D(3,1)=23   D(3,2)=-6   D(3,3)=12   D(3,4)=.00
```

Feldnamen unterliegen den gleichen Beschränkungen wie einfache Variablennamen; ein Feldname kann ein Buchstabe von A bis Z sein, dem eine Zahl von 0 bis 9 folgen kann (aber nicht muß). Die dem Feldnamen folgenden Klammern unterscheiden Feldvariablen von einfachen numerischen und Stringvariablen. Die maximale Größe eines Feldes wird durch den verfügbaren Speicherplatz bestimmt (siehe Anhang D, Seite 292). Nicht ganzzahlige Indices werden auf die nächste ganze Zahl gerundet; negative Indices sind unzulässig.

Setzen der Felduntergrenze (OPTION BASE)

Der HP-75 nimmt an, daß alle Feldindices bei 0 beginnen, sofern Sie nicht in einer OPTION BASE Anweisung eine andere Untergrenze spezifizieren.

```
OPTION BASE 1
            0
```

Diese Anweisung muß ausgeführt sein, bevor zum ersten Mal in einem Programm auf eine Feldvariable Bezug genommen wird. Mit OPTION BASE 1 setzt der Computer die Untergrenze aller Felder im momentanen Programm auf 1.

Jede OPTION BASE Anweisung ist eine *lokale* Deklaration und bezieht sich nur auf das gerade laufende Programm.

Felddeklarationen

Mit vier BASIC-Anweisungen können Sie die Art und die Größe von Feldern festlegen: DIM, REAL, SHORT und INTEGER. Bei der Initialisierung eines Programmes wird für Felder und andere Programmvariablen mit diesen Deklarationsanweisungen Speicherplatz bereitgestellt.

Ein Feld hat implizit den Typ REAL und die Obergrenze(n) 10, wenn es nicht explizit anders deklariert wurde. Wenn zum Beispiel das zweidimensionale Feld A(,) nicht in einer Deklarationsanweisung spezifiziert wird und das Programm die Einstellung OPTION BASE 0 setzt, wird A(,) bei der ersten Wertzuweisung auf 11x11 Werte mit REAL Genauigkeit dimensioniert, d.h. die Indices laufen von 0 bis 10.

Beachten Sie, daß eine Variable, falls notwendig, vor einer Wertzuweisung deklariert werden muß; andernfalls entsteht Fehler 35 – DIM exist var. Es ist üblich, Feldvariablen aus Dokumentationsgründen sehr früh im Programm zu deklarieren. Ein Programm kann mehr als eine Deklarationsanweisung enthalten, aber eine Variable kann in einem Programm nur einmal deklariert werden.

In IF...THEN Anweisungen sind die Anweisungen DIM, REAL, SHORT und INTEGER nicht erlaubt, und in Mehrfachanweisungs-Zeilen müssen sie als letzte Anweisung in der Zeile auftreten.

Felddimensionen (DIM)

DIM erlaubt die Zuordnung von Speicherplatz für numerische REAL Felder und für Stringvariablen.

```
DIM Variable [, Variable...]
```

Als zu dimensionierende Variable ist möglich:

- eine numerische Feldvariable, deren Feldobergrenzen in Klammern stehen müssen.
- eine Stringvariable, deren Anzahl Zeichen in Klammern angegeben ist. Wenn eine Stringvariable nicht deklariert wird, ist sie implizit zur Aufnahme von 32 Zeichen dimensioniert.

Beispiele:

```
20 OPTION BASE 1

30 DIM A(100)

40 DIM B(3,2),C$[96]
```

OPTION BASE Deklarationen müssen *vor* der Bezugnahme auf die erste Feldvariable im Programm erscheinen.
Deklariert ein eindimensionales Feld A() mit 100 Elementen: A(1) bis A(100).

Deklariert ein zweidimensionales Feld B(,) mit 6 Elementen (3 Zeilen x 2 Spalten) und einen String C$ mit maximal 96 Zeichen.

Die Anweisung DIM spezifiziert die Obergrenzen eines Feldes und die maximale Anzahl von Zeichen, die eine Stringvariable aufnehmen kann. Wenn für eine Feldvariable ein Index außerhalb des Definitionsbereichs spezifiziert wird, erhalten Sie Fehler 27 – invalid subscript; wenn Sie einer Stringvariablen zu viele Zeichen zuweisen, wird Fehler 42 – string too long – erzeugt.

Beachten Sie, daß die Indices in DIM Anweisungen als nichtnegative ganze Zahlen eingegeben werden müssen; andernfalls tritt Fehler 89 – bad parameter – auf.

Genauigkeitsdeklarationen (REAL, SHORT, INTEGER)

Für alle numerischen Variablen (einfachen und Feld-) wird volle Genauigkeit (REAL) unterstellt, wenn sie nicht in einer Genauigkeitsdeklaration anderweitig spezifiziert werden.

```
REAL numerische Variable [(Indices)] [, numerische Variable [(Indices)].....]
```

```
SHORT numerische Variable [(Indices)] [, numerische Variable [(Indices)].....]
```

```
INTEGER numerische Variable [(Indices)] [, numerische Variable [(Indices)].....]
```

Schlagen Sie die Deklaration einfacher numerischer Variablen auf Seite 80 in Abschnitt 5 nach. Die Anweisungen REAL, SHORT und INTEGER spezifizieren die Genauigkeit der einzelnen Elemente, wenn sie auf numerische Felder angewandt werden.

Beispiele:

```
10 OPTION BASE 0
20 INTEGER A,B,C(10)

30 SHORT P(20,25),P1,P2

40 REAL X5,D(4,4)
```

Deklariert die Untergrenze aller Felder im Programm.

Deklariert die Variablen A und B als INTEGER, und dimensioniert das Feld C() auf 11 ganzzahlige Elemente.

Deklariert und dimensioniert 546 das Feld P(,) auf (21 × 26) Elemente mit Genauigkeit SHORT; deklariert die Variablen P1 und P2 auf SHORT-Genauigkeit.

Deklariert das Feld D(,) und die Variable X5 auf REAL-Genauigkeit.

Beachten Sie, daß die Anweisung REAL hauptsächlich zur Programmdokumentation dient, da Feld- und einfache numerische Variablen auf REAL-Genauigkeit voreingestellt sind.

Wertzuweisung auf Felder (LET, INPUT, FOR...NEXT)

Den Elementen eines numerischen Feldes werden Werte auf die gleiche Weise zugewiesen wie bei einfachen Variablen: über das Tastenfeld oder innerhalb eines Programms. Auf die Elemente in M(,) oder A3() wird zum Beispiel mit M(1,2) bzw. A3(4) zugegriffen, und diesen Elementen können Werte zugewiesen werden. Die Wertzuweisung erfolgt gewöhnlich mit den Anweisungen LET und INPUT.

Beispiel:

```
>a3(4)=45
```

Mit RTN wird dem fünften Element des Feldes A3() der Wert 45 zugewiesen (bei OPTION BASE 0).

Dies ist eine über das Tastenfeld ausgeführte implizite LET-Anweisung. Bevor dem fünften Element ein Wert zugewiesen wird, wird die Variable A3() auf 11 (Indices 0 bis 10) Elemente der Genauigkeit REAL dimensioniert. Nach der Wertzuweisung kann das spezifizierte Element genau wie eine einfache numerische Variable benutzt werden.

Beispiel:

```
>a3(4)/5
```

```
9
```

Tastenfeldberechnung mit einem Element von A3().

Mit Hilfe von FOR...NEXT Schleifen können Sie den Zugriff auf Feldelemente rationalisieren.

Beispiel:

```
10 OPTION BASE 1
20 DIM A(5)
30 FOR I=1 TO 5
40 INPUT A(I)
50 NEXT I
60 FOR J=1 TO 5
70 DISP 'A(';J;')=';A(J)
80 NEXT J
```

Jedem Feldelement wird einzeln ein Wert zugewiesen.

Zeigt die Feldelemente an.

Zweidimensionale Felder werden oft mit Hilfe von verschachtelten FOR...NEXT Schleifen bearbeitet. Im folgenden Beispiel wird ein Integerfeld der Dimension 3x5 initialisiert, indem allen 15 Elementen der Wert 0 zugewiesen wird.

Beispiel:

```
10 OPTION BASE 1
20 INTEGER K(3,5)
30 FOR I=1 TO 3
40 FOR J=1 TO 5
50 K(I,J)=0
60 NEXT J
70 NEXT I
80 STOP
```

Verschachtelte FOR...NEXT Schleifen. Alle Elemente von Spalte 1 werden Null gesetzt, danach alle Elemente von Spalte 2 usw.

Die Anweisung STOP wurde gewählt, damit Sie die Werte des Feldes über das Tastenfeld kontrollieren können, bevor das Programm deallokatisiert wird und seine Werte verloren sind.

Beispiel:

```
>K(3,5)
```

```
0
```

Das spezifizierte Element soll angezeigt werden.

Mit DATA-Anweisungen können Sie Feldwerte innerhalb eines BASIC-Files speichern, so daß das Programm diese Werte direkt lesen und zuweisen kann (siehe auch Abschnitt 14, «Speichern und Abrufen von Daten»).

Stringausdrücke

Ein Stringausdruck oder Zeichenstring ist eine Gruppe von Zeichen, die als Einheit bearbeitet werden können. Die einfachste Form eines Stringausdrucks ist ein Text in Anführungszeichen. Diese bezeichnet man als String in Anführungszeichen oder *Literal*.

Stringausdrücke können die folgenden Formen annehmen:

- Literal; diese ist auf Zeichen beschränkt, die vom Tastenfeld aus angezeigt werden können (einschließlich Kommata und Leerzeichen, jedoch außer den Anführungszeichen, die den Ausdruck einschließen).
- Stringvariable; repräsentiert einen Speicherplatz, der Zeicheninformation enthält.
- Teilstring; ein Teil einer Stringvariablen.
- Stringfunktion; eine Operation, die Zeicheninformation zurückgibt.
- Beliebige Verkettungen der obenstehenden Stringausdrücke mit dem Operator &.

Wie ein numerischer Ausdruck kann auch ein String, wenn notwendig, in Klammern eingeschlossen werden.

Eine Stringvariable ist implizit auf eine Länge von 32 Zeichen dimensioniert, sofern ihre Länge nicht in einer DIM-Anweisung explizit spezifiziert wird.

Beispiel:

```
10 DIM A$[15], F$[28], H$[1000]
```

Dimensioniert A$ auf 15 Zeichen, F$ auf 28 Zeichen und H$ auf 1000 Zeichen.

Die Anzahl Zeichen der Stringvariablen muß in eckige (nicht runde) Klammern gesetzt werden.

Teilstrings

Teilstrings werden durch die Position ihres Anfangszeichens (oder ihres Anfangs- und Endzeichens) spezifiziert. Als Index ist jeder beliebige Zahlenausdruck erlaubt, dessen gerundeter Wert positiv ist und weder die dimensionierte noch die tatsächliche Länge des Strings überschreitet.

Mit der Form A$[I,I] können Sie ein einzelnes Zeichen innerhalb eines Strings spezifizieren. Den Rest eines Strings von einer bestimmten Position an spezifizieren Sie mit A$[I].

Beispiel:

```
200 A$[2,5]='Zusammen'

220 DISP A$[1,Y]

240 A$[20]=B$
```

Weist den Positionen 2 bis 5 von A$ die Zeichen usam zu.

Zeigt die Zeichen in A von Position 1 bis Position y an.

Weist dem Ende von A$ ab Position 20 die Zeichen in B$ zu.

Bearbeiten von Strings

Mit dem HP-75 können Sie Stringvariablen und Teilstrings auf die verschiedensten Weisen modifizieren. Mit Zuweisungsanweisungen werden einzelne Zeichen in Stringvariablen ersetzt.

Beispiel:

```
>C$='HOCH'@D$='TIEF'@DISP C$,D$
```

```
HOCH                    TIEF
```

Die Anfangswerte von C$ und D$.

```
>C$=D$@DISP C$,D$
```

```
TIEF                    TIEF
```

Der Inhalt von D$ wurde C$ zugewiesen.

Sobald einer Stringvariablen das erste Mal ein Anfangswert zugewiesen wird, stellt der HP-75 Speicherplatz für die dimensionierte oder voreingestellte Stringlänge zur Verfügung. Danach wird nur noch die aktuelle Stringlänge verwendet. Wenn Sie die Stringvariable umdefinieren, wird die momentane Länge auf den neuesten Stand gebracht, und die betroffenen Zeichen werden revidiert.

Bei der Umdefinition eines Teilstrings wird die momentane Länge gegebenenfalls erhöht, nur die betroffenen Zeichen werden revidiert. Sie können einen beliebigen Teil einer Stringvariablen durch einen anderen Stringausdruck oder einen Teil davon ersetzen. Spezifizieren Sie dazu die Indices der zu ändernden Zeichen der Variablen.

Beispiel:

```
>h$='Stock'
```

Weist H$ einen Anfangswert zu.

```
>h$[1,2]='Bl'
```

Ändert die Zeichen von H$ in Block.

Wenn an einen String angefügte Zeichen nicht direkt anschließen (einige dazwischenliegende Positionen bleiben unbestimmt), werden die Zwischenstellen mit Leerzeichen aufgefüllt.

Beispiel:

```
>H$[7,7]='B' @ DISP h$
```

```
Block B
```

Die Zeichen des Ersatzstrings werden von links nach rechts der Stringvariablen zugewiesen, bis die dimensionierte Größe der Variablen oder der spezifizierte Endindex erreicht ist.

Beachten Sie, daß der Nullstring spezifiziert wird, wenn der erste Index um 1 größer als der zweite Index ist. `H$[5,4]` zum Beispiel hat den selben Wert wie `''`. Wenn der zweite Index um mehr als 1 kleiner als der erste Index ist, erhalten Sie Fehler 42 – `string too long`.

Seien Sie beim Anfügen von Zeichen an einen String vorsichtig, wenn diese nicht direkt anschließen – es können noch immer zuvor definierte Zeichen vorhanden sein.

Beispiel:

```
 10 A$='123456789'
•20 DISP A$
 30 A$='X'
•40 DISP A$
 50 A$[5]='Y'
•60 DISP A$
```

Zeigt `123456789` an.

Zeigt `X` an.

Zeigt `X234Y` an. Die mittleren Zeichen in `A$` wurden nie geändert.

Sie können das Ende des Strings wie folgt löschen, um Verwirrung zu vermeiden:

```
25 A$[2]=''
```

Ersetzt alle Zeichen von Position 2 bis zum Stringende mit Nullzeichen; die Länge des Strings wird reduziert.

Wenn Sie jetzt das Programm starten, wird der String in Zeile 60 wie erwartet angezeigt:

```
X  Y
```

Stringfunktionen

Zur Kontrolle von Strings stehen Ihnen sieben Funktionen zur Verfügung. Die Argumente `S$` und `S1$` können Literale, Stringvariablen oder beliebige andere Stringausdrücke sein.

Funktion und Argument	Rückgabe
`LEN(S$)`	Derzeitige Länge von `S$`.
`POS(S$,S1$)`	Position von `S1$` in `S$`.
`VAL(S$)`	Der in `S$` dargestellte numerische Wert, wobei `S$` aus Ziffern, Dezimalpunkt und Exponenten bestehen kann.
`STR$(`*numer. Ausdruck*`)`	Der dem numerischen Ausdruck entsprechende Zeichenstring.
`UPRC$(C$)`	Das Äquivalent von `S$` in Großbuchstaben.
`KEY$`	Das der derzeit gedrückten Taste oder Tastenkombination zugeordnete Zeichen; oder der Nullstring, wenn keine Taste gedrückt wird.
`CAT$(`*Filenummer*`)`	Der Katalogeintrag des spezifizierten Files, oder der Nullstring, falls für die spezifizierte Filenummer kein Katalogeintrag besteht.

Argumente müssen in Klammern eingeschlossen werden. Technisch betrachtet sind LEN, POS und VAL numerische Funktionen, da sie Zahlenwerte zurückgeben. Sie werden an dieser Stelle behandelt, da sie vielfältige Stringmanipulationen ermöglichen.

Die Längenfunktion (LEN)

Die Funktion LEN gibt die derzeitige Anzahl von Zeichen in einem Stringausdruck zurück. Beachten Sie, daß eine Stringvariable nicht immer «gefüllt» ist; die Länge muß nicht immer der Dimensionierung entsprechen.

Beispiel:

```
>len('987')
```

```
3
```

Die Länge des Literals.

Beispiel: Schreiben Sie ein Programm, mit dem Sie einen Zeichenstring mit einer Länge von bis zu 40 Zeichen eingeben können. Benutzen Sie die Anweisung LEN, um einer Variablen diesen String in umgekehrter Zeichenfolge zuzuweisen. Wenn Sie zum Beispiel CAT eingegeben haben, sollte das Programm TAC anzeigen.

```
10 DIM W$[40],R$[40]

20 R$=''
30 INPUT 'Wort? ';W$
40 FOR I=LEN(W$) TO 1 STEP -1

50 R$=R$&W$[I,I]

60 NEXT I
70 DISP R$
```

Dimensioniert die Stringvariable auf eine Länge von maximal 40 Zeichen.

Initialisiert R$ als Nullstring.

Benutzt ein Literal als Eingabeaufforderung.

Benutzt die Wortlänge zur Schleifenzählung und zählt negativ.

Fügt durch Stringverkettung Zeichen in umgekehrter Reihenfolge an die Variable R$.

Ende der FOR...NEXT Schleife.

Zeigt das Umkehrwort an.

Versuchen Sie nach der Eingabe des Programms, ein paar Worte rückwärts zu buchstabieren!

```
Wort? 
                        PRGM
```

Geben Sie ein: international RTN.

```
lanoitanretni
```

Das Ergebnis bleibt in der Anzeige.

Nach der Modifikation eines Strings kann LEN unerwartete Ergebnisse liefern.

Beispiel:

```
 10 A$='AND'
•20 DISP LEN(A$)
 30 A$[3]='T'
•40 DISP A$
•50 DISP LEN(A$)
```

Zeigt 3 an.

Zeigt ANT an.
Zeigt 32, die Voreinstellung von A$, an.

Die Länge von A$ wurde auf 32 erhöht, da A$[3] ein Spezifikator ohne Endbestimmung ist. Dies passiert immer, wenn Sie einen Stringteil einschließlich des letzten Zeichens durch einen Ausdruck mit Spezifikator ohne Endwert ersetzen. Benutzen Sie einen Vollspezifikator, A$[3,3], um Verwirrung zu vermeiden.

Die Positionsfunktion (POS)

Die Funktion POS bestimmt die Position eines Strings innerhalb eines Strings.

Wenn der zweite String im ersten enthalten ist, gibt die Funktion POS die Stellung des ersten Zeichens des zweiten Strings innerhalb des ersten zurück. Wenn der zweite String nicht innerhalb des ersten enthalten ist, oder wenn der zweite String der Nullstring ist, wird der Wert 0 zurückgegeben. Tritt der zweite String mehrmals innerhalb des ersten auf, so gibt die Funktion nur die Position des ersten Auftretens zurück.

Beispiele:

```
>pos('ababc1234','123')
```

```
6
```

Sucht die Position des zweiten im ersten String.

Der zweite String beginnt bei der sechsten Stelle.

```
>a$='Versuche' @ b$='such'
```

Weist zwei Variablen Werte zu.

```
>pos(a$,b$)
```

```
4
```

Gibt den «Index» des Anfangs des Teilstrings zurück.

Mit Hilfe der Funktion POS können Sie Stringinformation in langen Stringvariablen einfügen und entfernen.

Umwandeln von Strings in Zahlen (VAL)

Normalerweise werden die Zeichen eines Strings nicht als numerische Daten erkannt und können in numerischen Berechnungen nicht verwendet werden. Sie wollen den String ja normalerweise exakt in der eingegebenen Form verwenden, Zeichen für Zeichen. Mit der Funktion VAL können Sie den Wert eines Zahlenstrings (Ziffern, Dezimalpunkt und Exponent) für Berechnungen verfügbar machen.

Beispiele:

```
>val('-3.3e12')
```

```
-3.3E12
```

Eingabe eines Literals.

Dies ist eine Zahl, und kein String, und kann einer numerischen Variablen zugewiesen werden.

```
100 INPUT A$ @ X=VAL(A$)
```

Die Anweisung INPUT akzeptiert einen String, der eine Zahl darstellen kann. Der Variablen X wird der entsprechende Zahlenwert zugewiesen.

Im Programm KONTO (Seite 156) wird auf ähnliche Weise Gebrauch von der Funktion VAL gemacht.

Das erste Zeichen des Stringarguments muß eine Ziffer, ein Plus- oder Minuszeichen, ein Dezimalpunkt oder ein Leerzeichen sein. Ein führendes Pluszeichen und führende Leerzeichen werden ignoriert; ein führendes Minuszeichen wird berücksichtigt. Die verbleibenden Zeichen im String oder Teilstring müssen Ziffern, ein Dezimalpunkt oder ein E sein. Ein E zwischen Ziffern wird als Exponentiationssymbol zur Basis 10 aufgefaßt.

Die von VAL zurückgegebene Zahl wird, abhängig von ihrer Größe, in Gleitkomma- oder Exponentialdarstellung ausgedrückt (Seite 73).

Das Stringargument kann mehr als eine Zahl enthalten. Alle aufeinanderfolgenden numerischen Zeichen werden als Teil der gleichen Zahl betrachtet, bis ein nichtnumerisches Zeichen im String auftritt.

Beispiel:

```
>val('3333rtrt6666')
```

```
3333
```

Die Funktion VAL wandelt den String in eine Zahl um, bis ein nichtnumerisches Zeichen gefunden wird.

Sie können einen anderen als den ersten numerischen Wert in einem String umwandeln, wenn Sie dessen Stellung spezifizieren. Wenn N$ zum Beispiel das obenstehende Literal enthält, kann der zweite Zahlenwert mit VAL(N$[9,12]) spezifiziert werden.

Umwandeln von Zahlen in Strings (STR$)

Die Funktion STR$ hat fast die Wirkung einer Umkehrfunktion zu VAL. Mit der Funktion STR$ können Sie eine Zahl in einen diese Zahl repräsentierenden String umwandeln.

Beispiel:

```
>v$=str$(120) @ disp v$&' Seiten'
```

```
120 Seiten
```

V$ enthält die Zeichen 120 und kann in anderen Stringausdrücken verwendet werden.

STR$ drückt die Zahl in Gleitkomma- oder Exponentialdarstellung (falls notendig) aus, einschließlich Minuszeichen, Dezimalpunkt und Exponenten. Beachten Sie, daß der String ohne führende oder nachlaufende Leerzeichen ausgegeben wird.

Umwandlung von Klein- in Großbuchstaben (UPRC$)

Die Funktion UPRC$ ermöglicht Ihnen, die Kleinbuchstaben in einem String in Großbuchstaben umzuwandeln.

Beispiel:

```
>m$='**"Zum Geburtstag"**'
```

Weist M$ eine Zeichenkombination zu.

```
>uprc$(m$)
```

```
**"ZUM GEBURTSTAG"**
```

Alle Kleinbuchstaben wurden umgewandelt.

Kleinbuchstaben haben andere Dezimalcodes als Großbuchstaben. Mit Hilfe der Funktion UPRC$ können Sie Strings ohne Berücksichtigung von Klein- oder Großschreibung vergleichen. Ein Teil eines Programms könnte zum Beispiel so aussehen:

```
:
30 INPUT A$
40 IF UPRC$(A$[1,1])='J' THEN
:
```

Der Benutzer kann J, j, ja, JA usw. eingeben, so daß das Programm zu Anweisung 80 verzweigt.

Beachten Sie, daß UPRC$ nicht auf unterstrichene Kleinbuchstaben (Dezimalcodes 225 bis 250) anspricht.

Kontrolle von Programmen über das Tastenfeld (KEY$)

Das Tastenfeld ist zwar während der Programmausführung normalerweise desaktiviert, der HP-75 überprüft aber dennoch kontinuierlich, ob Tasten gedrückt werden. Die Funktion KEY$ gibt das einer beliebigen gedrückten Taste zugeordnete Zeichen zurück und ermöglicht damit über das Tastenfeld gesteuerte Verzweigungen.

KEY$ gibt ein einzelnes Informationszeichen zurück. Wenn in dem Moment, wo der HP-75 das Tastenfeld abfragt, keine Taste gedrückt ist, ergibt KEY$ den Nullstring.

Sie können das Programm eine «Leerschleife» ausführen lassen, in der der HP-75 dann auf das Drücken einer bestimmten Taste wartet.

```
90 WIDTH 32
100 K$=KEY$ @ IF K$='' THEN 100


100 DISP 'Zeichen= ';K$;' Code
=';NUM(K$);
120 IF K$="T" THEN DISP TIME$
130 IF K$=CHR$(136) THEN DISP DA
TE$
140 GOTO 100
```

Wenn zu Beginn dieser Zeile noch keine Taste gedrückt wurde, wird K$ zum Nullstring, und die Ausführung beginnt wieder am Zeilenanfang.

Zeigt das Zeichen und den Dezimalcode der gedrückten Taste an.

Prüft auf eine bestimmte Tastenfolge, SHIFT T.

Prüft auf die Taste I/R.

Rücksprung zur Warteschleife.

Starten Sie das Programm mit RUN. Drücken Sie danach das nicht umgeschaltete A:

```
Zeichen= a Code= 97
```

Zeigt das Zeichen und den Dezimalcode der nicht umgeschalteten Taste A an.

Versuchen Sie nun die Taste I/R:

```
Zeichen= _ Code=136 02/15/83
```

Das Datum wird wie spezifiziert angezeigt. Mit der Anweisung IF...THEN können Sie feststellen, welche Taste gedrückt wurde.

Mit der Funktion KEY$ ist eine Vielzahl von Tastenfeldeingriffen möglich. Sie können mit KEY$ jede Programmieroperation auslösen, beispielsweise auf Tastendruck auf eine vollständig neue Routine verzweigen. Die Programme AUFGPST (Seite 170) und NAMEN (Seite 213), die dem Benutzerpaket beiliegen, machen ausführlich Gebrauch von Verzweigungen mit KEY$.

Beachten Sie, daß Sie die Taste ATTN möglicherweise zweimal hintereinander drücken müssen, um die Warteschleife zu unterbrechen. Beim ersten Drücken kann ATTN als gewöhnliche Taste verstanden werden; die Funktion KEY$ gibt dann △, Dezimalcode 128, zurück, anstatt das Programm zu unterbrechen.

Die Katalogfunktion (CAT$)

Mit Hilfe der Funktion CAT$ kann ein Programm auf den gesamten Systemkatalog zugreifen und Namen, Typ, Größe, Zeit und Datum von Files zurückgeben. CAT benutzt ein numerisches Argument und gibt 32-Zeichen Strings zurück.

Files werden nach ihrer Reihenfolge im Speicher spezifiziert. Je höher das Argument, desto älter ist der File. CAT$(1) spezifiziert zum Beispiel den zuletzt erzeugten File.

CAT$(0) spezifiziert den momentanen EDIT-File. Dieser kann, aber muß nicht, mit CAT$(1), dem neuesten File im Speicher, identisch sein.

CAT$ einer negativen Zahl spezifiziert den File, auf den der Programmpointer zur Zeit gerichtet ist.

CAT$ rundet Argumente auf die nächsthöheren ganzzahligen Werte. Ist ein Argument größer als die Anzahl der Files im Speicher, erzeugt CAT$ dieses Arguments den Nullstring.

Beispiel: Schreiben Sie ein Programm PRINTALL, das den Katalogeintrag und danach ein Listing jedes Files im Speicher, vom neuesten bis zum ältesten, ausgibt. Wenn die Files appt und keys existieren, sollen nur deren Katalogeinträge ausgedruckt werden. Erzeugen Sie zuerst den Programmfile, indem Sie edit 'printall' RTN eingeben. Tasten Sie dann ein:

```
 10 PWIDTH 32
•20 N=1
•30 C$=CAT$(N)
•40 IF C$='' THEN STOP

•50 IF C$[1,4]='appt' OR C$[1,4]=
 'keys' THEN PRINT C$ @ GOTO 80
•60 PRINT C$ @ PLIST C$

•70 PRINT
 80 N=N+1 @ GOTO 30
```

Spezifikation des neuesten Files im Speicher.

Füllt C$ mit den Zeichen eines Katalogeintrags.

Stoppt das Programm, wenn der letzte Fileeintrag gelesen ist.

Prüft auf die Files appt und keys, und druckt nur deren Katalogeintrag.

Druckt den Katalogeintrag eines anderen Files und druckt den File aus.

Druckt eine Leerzeile zwischen zwei Files.

Wenn die Anzeige des HP-75 die derzeitige PRINTER IS Einheit ist, und wenn PRINTALL der neueste File im Speicher ist, ist der File PRINTALL der erste File, der ausgedruckt wird. Drücken Sie RUN:

```
PRINTALL B 168 09:40 02/15/83
10 PWIDTH 32
20 N=1
⋮
```

Der Katalogeintrag von PRINTALL zeigt den initialisierten Zustand; es ist das derzeit laufende Programm.
Und so weiter, durch den gesamten Speicher.

Jeder Katalogeintrag enthält eine Anzahl von zwischen Filenamen und Filetyp, Filetyp und Filelänge usw. eingebetteten Leerzeichen. Da das erste Leerzeichen nach einem Filenamen den Spezifikator des Filenamens abschließt, kann der ganze Katalogeintragsstring (im Beispiel C$) als Parameter für den Befehl PLIST und andere Filebefehle wie PURGE und RENAME benutzt werden.

> **Hinweis:** Die Namen der Files workfile, appt und keys können nicht mit Stringvariablen in Filebefehlen spezifiziert werden. Wenn F$ zum Beispiel der Wert keys zugewiesen wird, sucht der HP-75 nach PLIST F$ den Speicher nach einem Benutzerfile namens KEYS, und nicht nach dem Systemfile keys ab.

Vergleiche von Stringausdrücken

Mit den Vergleichsoperatoren =, >, >=, <, <=, <> und # können Sie auch Zeichenstrings vergleichen.

Beispiel:

```
'A'<'B'
```

```
 1
```

Der Ausdruck ist «wahr» und erhält deshalb den Wert 1.

Als Vergleichsgrundlage dient im obigen Beispiel der Dezimalcode des Zeichens A (65) und der Dezimalcode von B (66).

Zwei Strings werden als gleich betrachtet, wenn sie genau die gleichen Zeichen in der gleichen Reihenfolge enthalten. Die Strings werden dabei Zeichen für Zeichen miteinander verglichen, bis ein Unterschied gefunden wird. Wenn ein String endet, bevor ein Unterschied gefunden wurde, wird der kürzere String als der kleinere betrachtet. Wenn ein Unterschied gefunden wurde, wird der kleinere anhand der Dezimalcodes *der zwei verschiedenen Zeichen* ermittelt.

Beispiel:

```
'AAbcde'<'ABb'
```

```
 1
```

Das zweite Zeichen (A gegen B) bestimmt das Ergebnis des Vergleichs.

Die Anweisung PUT

Mit der Anweisung PUT können Sie innerhalb eines Programms das Drücken von Tasten simulieren.

```
PUT Stringausdruck mit einem Zeichen
```

Der HP-75 verfügt über einen «Wartetasten»-Buffer, der das zur momentan gedrückten Taste gehörende Anzeigezeichen abspeichert. Geben Sie zum Beispiel `wait 10` RTN ein, um den HP-75 10 Sekunden warten zu lassen, und drücken Sie dann die Tasten Q, W, E, R, T und Y, solange er wartet.

```
>q
```

Die Anzeige zeigt, daß nur das erste Zeichen der Folge im Wartetasten-Buffer gespeichert worden ist.

Wenn Sie zum Beispiel das Tastenfeld mit SHIFT LOCK auf Großbuchstaben umschalten, wird das Zeichen mit dem Dezimalcode 172 in den Wartetasten-Buffer eingegeben.

Mit der Anweisung PUT kann ein *Programm* ein Zeichen in den Wartetasten-Buffer ablegen. Die Wirkung ist die gleiche, als wenn die Taste tatsächlich gedrückt worden wäre.

Beispiele:

```
10 PUT '↵'


20 INPUT 'Ihr Name: ';A$
⋮
990 PUT CHR$(129)

1000 END
```

Das Zeichen ↵ wird erzeugt, indem Sie SHIFT I/R und dann SHIFT LOCK drücken. Diese Anweisung speichert die Tastenfolge SHIFT LOCK im Wartetasten-Buffer

Das Tastenfeld ist auf Großbuchstaben umgeschaltet.

Legt ▫ im Wartetasten-Buffer ab, wie beim Drücken von TIME.

Der HP-75 wird in TIME-Modus geschaltet.

Beachten Sie, daß Sie eine Taste oder Tastenkombination mit ihrem Dezimalcode und der Funktion CHR$ spezifizieren können, wie in Zeile 990 geschehen. Mit dieser Möglichkeit können Sie unerwartete Ergebnisse beim Auslisten des Programms auf einem externen Drucker vermeiden.

Der Wartetasten-Buffer kann immer nur *ein* Zeichen aufnehmen. Der Tastencode im Warte-Buffer geht verloren, sobald eine Ausgabeoperation wie DISP, LIST oder CAT eine End-Of-Line Sequenz erzeugt. Deshalb sollte die Anweisung PUT einer Anweisung, die das Tastenfeld aktiviert, wie zum Beispiel INPUT, STOP oder END, unmittelbar vorausgehen. Andernfalls wird die Anweisung PUT ignoriert, analog wie während der Programmausführung gedrückte Tasten. Beachten Sie, daß die Programmausführung unterbrochen wird, wenn Sie die Taste ATTN (Dezimalcode 128) in der Anweisung PUT spezifizieren, genau wie wenn Sie die Taste ATTN drücken.

Mit der Anweisung PUT wird auf die *derzeitige* Definition der spezifizierten Taste zugegriffen. Wenn SHIFT Z zum Beispiel mit der Anweisung PRINT belegt wurde, bewirkt PUT 'Z' die Ausführung der Anweisung PRINT.

Simulation von Stringfeldern

Die Teilstringfähigkeiten des HP-75 ermöglichen die Simulation von eindimensionalen Stringfeldvariablen. Eine Möglichkeit besteht darin, eine große Stringvariable als Feld selbst zu benutzen und in Teilstrings von gleicher Länge zu unterteilen, die die Elemente des Stringfeldes darstellen.

Beispiel:

```
•10 DIM A$[200]
 20 FOR I=1 TO 5
 30 DISP 'Stringelement Nr.';I;
•40 INPUT A$[I*25-24,I*25]


 50 NEXT I
 60 FOR I=5 TO 1 STEP -1
•70 DISP I;A$[(I-1)*25+1,I*25]
 80 NEXT I
```

Dimensioniert die Stringvariable.

Das erste Element (I = 1) belegt belegt die Zeichenpositionen 1 bis 25; das zweite Element die Positionen 26 bis 50 usw.

Zeigt den Index und den Wert jedes «Feldelements» an.

Die Wahl von 25 als Faktor für den Stringindex legt die Länge jedes Teilstrings auf 25 Zeichen fest.

Bei der Ausführung dieses Programmes werden fünf Stringelemente angenommen und den «Elementen» von A$ zugewiesen. Nach der ersten FOR...NEXT Schleife werden die Elemente in der zweiten FOR...NEXT Schleife in umgekehrter Reihenfolge angezeigt.

Sie können auch mit der Funktion POS die Indices von Teilstrings innerhalb simulierter Stringfelder auffinden.

Benutzerdefinierte Funktionen

Auf dem HP-75 können Sie innerhalb eines Programmes Ihre eingenen Funktionen definieren und bei der Ausführung des Programms wie Systemfunktionen benutzen. Benutzerdefinierte Funktionen können auf null bis 36 Argumenten oder *Parametern* operieren und geben einzelne Werte zurück. *Numerische* Funktionen geben Zahlenwerte zurück, *Stringfunktionen* geben Stringwerte zurück.

Einzeilige Definitionen (DEF FN)

Die Anweisung DEF FN wird in allen Funktionsdefinitionen als die erste Anweisung benötigt. In einer einzeiligen Definition kann sie allein vorkommen:

```
DEF FN Name einer numerischen Variablen [(Parameter [,Parameter...])] =numerischer Ausdruck
DEF FN Name einer Stringvariablen [(Parameter [,Parameter...])] =Stringausdruck
```

Der Name einer numerischen Funktion besteht aus den Buchstaben FN, gefolgt von dem Namen einer numerischen Variablen (zum Beispiel FNS und FNA3). Bei einer Stringfunktion folgt auf FN der Name einer Stringvariablen (zum Beispiel FNZ$ oder FNZ1$). Die Parameterliste kann eine beliebige Kombination von einfachen numerischen oder Stringvariablen sein, die durch Kommata getrennt sind. Feldnamen sind nicht erlaubt.

Beispiele:

```
20 DEF FNR(X,Y) = SQR(X^2+Y^2)
30 DEF FNH(D) = (EXP(1)^D+EXP(1
)^(-D))/2
40 DEF FNS$(I) =A$[I*5-4,I*5]
50 DEF FNE$(C,R) = CHR$(27)&'%'&
CHR$(C)&CHR$(R)
```

Berechnet $\sqrt{x^2+y^2}$.
Berechnet den Cosinus Hyperbolicus des Parameters.

Gibt einen Teilstring des Strings A$ zurück.
Gibt einen Escapecode zur Positionierung des Cursors zurück.

Eine Funktionsdefinition kann an beliebiger Stelle, vor oder nach Bezugnahme auf die Funktion, im Programm erscheinen. Es ist jedoch üblich, Funktionsdefinitionen an den Anfang von Hauptprogrammen zu stellen.

Im folgenden Programmsegment werden die oben definierten Funktionen aufgerufen:

```
•60 A,B=30 @ C=FNR(A,B)

 70 DISP A;B;C
 80 X=1 @ Z=FNH(X)
 90 DISP X;Z
•100 A$='1234567890'

 110 DISP FNS$(1)
 120 DISP FNE$(16,15);'HIER'
```

Die Funktion erhält beim Aufruf die Programmkontrolle und die benötigten Parameterwerte.

A$ ist innerhalb der Funktion FNS$ verfügbar, obwohl es eine Variable des Hauptprogramms ist.

Bei der Ausführung zeigt das Programm die folgenden vier Zeilen an:

```
 30   30   42.4264068712
 1   1.54308063482
12345
                HIER
```

Wurzel der Quadratsumme.
Cosh (1).
Die ersten fünf Zeichen in A$.
Der spezifizierte String, von Spalte 16 an.

Das Hauptprogramm kann die Werte von Konstanten, Variablen, Feldelementen und Ausdrücken durch die Parameterliste im Hauptprogramm an benutzerdefinierte Funktionen weitergeben. Weiterhin sind alle Variablen des Hauptprogramms *global*, das heißt, für alle benutzerdefinierten Funktionen verfügbar. Die Stringvariable A$ des Hauptprogramms wird zum Beispiel von FNS$ direkt abgerufen.

Einschränkungen:

- Benutzerdefinierte Funktionen können nicht über das Tastenfeld definiert oder aufgerufen werden.
- Variablen, die nur in der Parameterliste der Funktionsdefinition erscheinen, sind *lokal* auf die Funktion beschränkt und bleiben für das Hauptprogramm unbekannt. Auf Parametervariablen kann nur mit Instruktionen zugegriffen werden, die zwischen den Zeilen DEF FN und END FN angeordnet sind.
- Die Anweisung DEF FN muß die erste und einzige Anweisung innerhalb der Programmzeile sein.
- Die Länge eines Stringparameters, der zwischen Hauptprogramm und Funktion übergeben wird, ist auf 32 Zeichen voreingestellt. Sie können aber in der Parameterliste der Anweisung DEF FN längere Strings deklarieren (Seite 209).
- Mehrzeilige Stringfunktionen geben Strings mit 32 Zeichen zurück. Bei längeren Strings tritt Fehler 42 – string too long – auf.
- Einzeilige Stringfunktionen können Strings beliebiger Länge, sofern sie in eine Zeile passen, zurückgeben.
- Die Anzahl von Parametern, die eine Funktion akzeptieren kann, ist durch die Länge der Zeile mit der Anweisung DEF FN beschränkt. Die obere Grenze beträgt 36 Parameter.

- Eine Funktion darf nicht *rekursiv* definiert werden; das heißt, der Name der Funktion darf nicht rechts vom Gleichheitszeichen in der Definitionsgleichung auftreten.
- Jede Funktion muß einen eindeutigen Namen besitzen.

Eine Funktion muß kein Argument besitzen, um einen Wert zurückzugeben. (Denken Sie an die Funktionen PI, EPS und INF.)

Beispiele:

```
1000 DEF FNW$='Bericht zur
Betriebssicherheit'
1010 DEF FNP=6.625E-27
```

FNW$ gibt eine Stringkonstante zurück.

FNP gibt die Planck'sche Konstante zurück.

Mehrzeilige Funktionen (DEF FN, LET FN, END DEF)

Sie können mit einem Block, oder einer Reihe von Programmanweisungen auch anspruchsvollere Funktionen definieren. Die Definition einer mehrzeiligen Funktion umfaßt drei grundlegende Elemente. Die erste Anweisung muß die Anweisung DEF FN sein.

```
DEF FN Name einer numerischen Variablen [(Parameter [,Parameter...])]
DEF FN Name einer Stringvariablen [(Parameter [,Parameter...])]
```

Beachten Sie, daß der Funktion in dieser Form der Anweisung DEF FN kein Wert zugewiesen wird. Es sollte aber mindestens eine der Anweisungen in der Definition der Funktion dem Funktionsnamen einen Wert zuweisen.

```
[LET] FN Name einer numerischen Variablen = numerischer Ausdruck
[LET] FN Name einer Stringvariablen = Stringausdruck
```

Beachten Sie, daß das Schlüsselwort LET in der Anweisung optional ist, und daß die Parameterliste wegfällt. Die Zuweisungsanweisung überträgt den berechneten Wert der Funktion auf den Funktionsnamen im Hauptprogramm. Es dürfen in einer Funktionsdefinition mehrere Zuweisungsanweisungen auftreten.

Die letzte Anweisung einer Mehrzeilenfunktion muß die Anweisung END DEF (*end definition*) sein.

```
END DEF
```

Die Anweisung END DEF muß als letzte Anweisung in einer Programmzeile auftreten und ist nach THEN nicht gestattet.

Zwischen DEF FN und END DEF können eine beliebige Anzahl weiterer Anweisungen stehen. Wenigstens eine dieser Anweisungen sollte dem Funktionsnnamen das Ergebnis der Funktion zuweisen.

Das Programm QUADGLCH

Das folgende Programm QUADGLCH berechnet die reellen Lösungen einer quadratischen Gleichung der Form:

$$ax^2 + bx^2 + c = 0.$$

Die Lösung hat die Form $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$, wobei $a \neq 0$ und $b^2 \geqslant 4ac$.

```
 10 ! Quadratische Gleichungen
•20 DEF FNQ(X,Y,Z)
 30 IF X=0 OR Y^2<4*X*Z THEN FNQ=
 -1 ELSE FNQ=SQR(Y^2-4*X*Z)
•40 END DEF
 50 ! Anfang QUADGLCH
 60 INPUT 'Eingabe a,b,c: ';A,B,C
 70 IF FNQ(A,B,C)=-1 THEN DISP 'B
 itte neuer Versuch.' @ STOP
•80 R1=(-B+FNQ(A,B,C))/(2*A)
•90 R2=(-B-FNQ(A,B,C))/(2*A)
 100 DISP 'Loesung 1= ';R1
 110 DISP 'Loesung 2= ';R2 ';'Pro
 be:'
 120 DISP 'ax^2+bx+c=';
•130 DISP A*R1^2+B*R1+C
•140 DISP 'und';A*R2^2+B*R2+C
 150 END
```

Die Funktion prüft auf Lösbarkeit und berechnet dann $\sqrt{b^2-4ac}$.

Mit FNQ werden die Lösungen 1 und 2 berechnet.

Prüft die Lösungen der Originalgleichung.

Geben Sie beim Start des Programms passende Werte ein. Beispiel:

```
Eingabe a,b,c: ▮
                    PRGM
```

Geben Sie 2, 3, −1 RTN ein.

Das Programm zeigt mit der momentanen Verzögerungsrate vier Zeilen an.

```
Loesung 1= .280776406405
Loesung 2=-1.78077640641 . Probe:
ax^2 + bx + c= 0
und .000000000003
```

R1 und R2 sind gute Näherungen.

Weitere Betrachtungen

Der Zweck von benutzerdefinierten Funktionen ist, mit der Funktion Parameter-Werte des Hauptprogramms zu einem einzelnen Wert zu kombinieren; es ist am besten, benutzerdefinierte Funktionen auf diese Aufgabenstellung zu beschränken. Wenn eine Funktion zum Beispiel die Anweisung PRINT oder DISP enthält und in einer PRINT oder DISP Anweisung des Hauptprogramms selbst erscheint, erhalten Sie möglicherweise nicht die erwartete Ausgabe. Wenn Sie in oder aus einer Funktionsdefinition verzweigen, können Sie unerwartete Fehler erzeugen.

Es hat sich bewährt, für Funktionsparameter andere Variablennamen zu verwenden als für die Variablen des Hauptprogramms, damit Sie diese nicht miteinander verwechseln.

Alle *Variablen* des Hauptprogramms sind für benutzerdefinierte Funktionen erreichbar; das heißt eine Funktion kann beliebige Variablen des Hauptprogramms aufrufen und benutzen. Wenn dieser Wert jedoch geändert wird, bleibt die Änderung auch im Hauptprogramm erhalten; Sie sollten sich über mögliche Nebeneffekte im klaren sein.

Die Parameter einer benutzerdefinierten Funktion können über das Tastenfeld, anders als Tastenfeldvariablen, nicht erreicht oder geändert werden. Sie können einem Funktionsparameter zum Beispiel keinen anderen Wert zuweisen, wenn das Programm mit der Anweisung STOP angehalten wurde.

Die Länge von an eine Funktion weiterzugebenden Stringparametern ist auf 32 Zeichen voreingestellt. Sie können in der Funktionsdefinition längere Strings spezifizieren, indem Sie die Länge in Klammern an den Stringnamen anfügen. Zum Beispiel:

```
4000 DEF FNS$(A$,B$[8],C$[96])
```

Stellt Speicherplatz für Stringparameter mit 32 (A$), 8 (B$) und 96 (C$) Zeichen bereit.

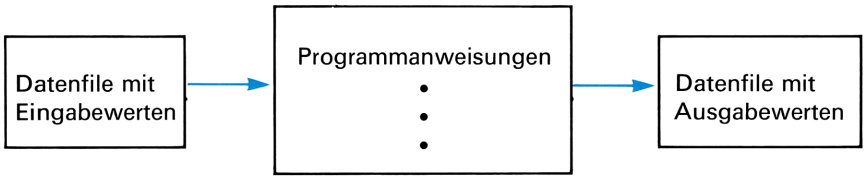
Abschnitt 14

Speichern und Abrufen von Daten

Inhalt

Einführung

Die interne Filestruktur des HP-75 ermöglicht Ihnen eine Vielzahl von Eingabe- und Ausgabeoperationen. Ein Programm kann zum Beispiel mit den Werten eines Eingabefiles arbeiten und die Ergebnisse in einen Ausgabefile schreiben.



Sie können in jedem BASIC-File Datenwerte, gemischt mit anderen Programmanweisungen, speichern; wir wollen im folgenden einen BASIC- File, der *nur* Daten enthält, als *Datenfile* bezeichnen. In diesem Abschnitt werden Sie sehen, wie mit den Anweisungen DATA, READ und RESTORE innerhalb eines Programmes Daten gespeichert und wieder abgerufen werden können, und wie *Datenfiles* erzeugt und mit den Anweisungen ASSIGN #, PRINT #, READ # und RESTORE # angesprochen werden können. Die Anweisung DATA speichert Zahlen- und Zeicheninformation für den späteren Gebrauch. Mit der Anweisung PRINT # können Sie DATA Anweisungen in anderen BASIC- und Textfiles im Speicher erzeugen. Mit den Anweisungen RESTORE und RESTORE # kann ein Programm das Schreiben und Lesen von Daten in Programm- und Datenfiles kontrollieren.

Lesen von in Programmen abgelegten Daten (DATA, READ)

Die Anweisung DATA enthält eine Liste von Zahlen und Zeichen, die in einem BASIC-File gespeichert werden sollen.

DATA *Zahl oder Text* [, *Zahl oder Text....*]

Die Datenelemente müssen durch Kommata getrennt werden. Zwei aufeinanderfolgende Kommata in der Anweisung DATA stehen für den Nullstring.

Beispiele:

```
100 DATA 1900,2100,2200
```

Datenelemente können Zahlen,

```
110 DATA funktional,'Vertrag'
```

Literale mit oder ohne Anführungszeichen (mit ' ' oder " " als Anführungszeichen)

```
120 DATA 1900,funktional,2200,,''
```

oder eine Kombination daraus sein. Die letzten beiden Datenelemente sind Nullstrings.

Jede DATA Anweisung kann eine Anzeigezeile mit Information – bis zu 96 Zeichen, einschließlich Zeilennummer, Leerzeichen, Schlüsselwort DATA, Datenelementen, Kommata und zwei für Eingabeaufforderung und Cursor reservierter Positionen, aufnehmen (siehe auch «Lange Datenzeilen», Seite 228). Die Werte in der DATA Anweisung werden im BASIC-File gespeichert, bis die Anweisung geändert oder gelöscht wird, oder der BASIC-File gelöscht wird. Beachten Sie, daß Datenelemente *Konstanten* sein müssen. Variablennamen oder Schlüsselworte werden als einfacher Text interpretiert. Numerische Konstanten können ein führendes Plus- oder Minuszeichen und einen Dezimalpunkt enthalten und können in Exponentialdarstellung mit Exponenten E ausgedrückt werden, zum Beispiel −2.819E−29. Sehr große und sehr kleine Zahlen werden in einer DATA Anweisung beim Drücken von RTN in Exponentialdarstellung umgewandelt.

Text *ohne* Anführungszeichen unterliegt einigen Beschränkungen:

- Führende und nachlaufende Leerstellen werden ignoriert.
- Kommata im Text werden als Trennzeichen für Datenelemente interpretiert.
- Ein Parameter wird als Zahl interpretiert, wenn die erste Stelle eine Ziffer oder ein + oder − ist.

Diese Beschränkungen gelten nicht für Text in Anführungszeichen.

Beispiel:

```
122 DATA '2b,',' od. ',nicht,' 2b'
```

Benutzen Sie Anführungszeichen für Strings mit führenden Ziffern, Kommata, oder führenden und nachlaufenden Leerzeichen.

Nach THEN oder ELSE dürfen keine DATA Anweisungen stehen. Die Anweisung DATA sollte auch nicht in Zeilen mit Mehrfachanweisungen stehen, da das Symbol @ nach dem Schlüsselwort DATA als gewöhnliches Textzeichen und nicht als Verknüpfungssymbol interpretiert wird. Wenn die Anweisung DATA auf eine andere Anweisung in der gleichen Zeile folgt, kann der HP-75 auf die Daten nicht zugreifen. Wenn die DATA Anweisung über das Tastenfeld eingegeben wird, meldet der HP-75 Fehler 88 – bad statement.

Die Anweisung READ spezifiziert Variablen, denen Werte aus den DATA Anweisungen innerhalb des Programms zugewiesen werden sollen.

READ *Name der Variablen* [, *Name der Variablen...*]

Die READ Liste kann aus einfachen numerischen Variablen, numerischen Feldelementen, Stringvariablen und Teilstrings bestehen, die durch Kommata getrennt sein müssen.

Beispiel:

```
210 READ A,B(3),C$,D$[5]
```

Diese Anweisung weist zwei numerischen und zwei Stringvariablen Werte zu.

Sie können mit einer einzelnen Anweisung READ ein ganzes Feld einlesen. Lassen Sie einfach die Indices in den Klammern weg, das Komma in zweidimensionalen Feldern sollte jedoch stehen bleiben. Mit READ A(), B(,) zum Beispiel werden alle Elemente des eindimensionalen Feldes A und des zweidimensionalen Feldes B eingelesen.

Das Betriebssystem arbeitet mit einem speziellen Filemechanismus, dem sogenannten *Datenpointer*, um das Datenelement zu kennzeichnen, das als nächstes eingelesen werden soll. Das erste Datenelement in der DATA Anweisung mit der niedrigsten Zeilennummer wird zuerst gelesen. Danach springt der Datenpointer jedesmal um ein Datenelement nach rechts, wenn ein neuer Wert gelesen wurde. Nach dem letzten Wert in einer DATA-Anweisung zeigt der Datenpointer auf den ersten Wert der nächsthöher numerierten DATA Anweisung und so fort.

Beispiel:

```
10 ! Programmanfang
 ↑
 a
⋮
400 DATA 4
         ↑
         b
410 ! Mitte des Programms
⋮
600 DATA 9, 1,8, 4,7,9
         ↑    ↑
         c    d
610 ! Programmende
⋮
```

a. Der Datenpointer des Programms steht zu Beginn auf der ersten Zeile des Files.

b. Mit READ X wird das erste Datenelement des Files, 4, gelesen; der Datenpointer bleibt dahinter stehen.

c Mit READ Y wird das nächste Datenelement im File, 9, gelesen; der Datenpointer bleibt dahinter stehen.

d. Mit READ Z,T werden die nächsten zwei Datenelemente, 1 und 8, gelesen; der Datenpointer wird wieder umpositioniert usw.

Der wesentliche Punkt ist, daß das Einlesen der Daten durch die Reihenfolge der Datenelemente von links nach rechts und von kleinster Zeilennummer zu höherer Zeilennummer bestimmt wird.

```
400 DATA 4,9,1,8,4,7,9
```

```
400 DATA 4
600 DATA 9,1,8,4,7,9
```

```
400 DATA 4
600 DATA 9,1,8
800 DATA 4,7
803 DATA 9
```

Diese drei Datengruppen werden auf gleiche Weise sequentiell (oder *seriell*) gelesen.

Wenn ein Programm versucht, mehr Daten zu lesen, als Datenelemente vorhanden sind, tritt bei der READ Anweisung Fehler 34 – no data – auf. Die Anweisung READ kann über das Tastenfeld nicht ausgeführt werden.

Für jede numerische Variable in einer READ Anweisung muß eine numerischer Konstante in einer DATA-Anweisung vorhanden sein. Die Genauigkeit eines DATA Zahlenwerts wird durch die Genauigkeit der Variablen im Programm (REAL, SHORT oder INTEGER) bestimmt, und bei der Wertzuweisung an die Variable wird gerundet (wenn erforderlich). Stringvariablen in der READ Liste müssen richtig dimensioniert sein, um alle in einer DATA Anweisung gelesenen Zeichen aufnehmen zu können, andernfalls entsteht Fehler 42 – string too long, und es findet keine Zuweisung statt. Wenn eine Zahl (zum Beispiel 5.55) in eine Stringvariable eingelesen wird, wird diese als Zeicheninformation und nicht als Zahleninformation behandelt.

DATA Anweisungen können vor und nach die READ Anweisung gestellt werden. In unseren Beispielen sind DATA Anweisungen an das Ende von Programmen gestellt. Während der Programmausführung werden DATA Anweisungen ignoriert, wenn keine dazugehörigen READ Anweisungen vorliegen.

Beispiel: Das folgende Programm liest Zahlen aus einer DATA Anweisung sequentiell ein und berechnet deren Quadrate:

```
10 READ A
20 FOR I=1 TO A

30 READ X

40 DISP X; 'quadrat=';X^2


50 NEXT I
60 DATA 4,1,1,2,3,5,8,13
```

Liest die erste Zahl der Datenliste in Zeile 60.

Der Wert von A (4) bestimmt die Anzahl Schleifendurchläufe.

Sequentielles Lesen von Daten in der DATA Anweisung.

Variable X erhält nacheinander die Werte der Datenelemente.

Vier Datenelemente werden in der Schleife gelesen.

Die verbleibenden Datenelemente (5, 8 und 13) werden nicht benutzt.

Drücken Sie [RUN]:

```
1 quadrat=1
1 quadrat=1
2 quadrat=4
3 quadrat=9
```

Mit dem Wert des ersten Datenelements kann die Anzahl der Schleifendurchläufe geändert werden. Mit FOR...NEXT Schleifen können Sie sehr effizient auf DATA Werte zugreifen.

Nochmaliges Lesen von DATA Anweisungen (RESTORE)

Mit der Anweisung RESTORE wird der Datenpointer entweder auf die spezifizierte Zeile des Files oder auf die DATA Anweisung mit der niedrigsten Nummer (wenn keine Zeile spezifiziert wurde) positioniert.

```
RESTORE [Zeilennummer]
```

Die *Zeilennummer* muß eine vorzeichenlose ganze Zahl von 0 bis 9999 sein. Die spezifizierte Zeile muß mit dem Schlüsselwort DATA beginnen, braucht aber keine weiteren Informationen zu enthalten.

Beispiel:

```
 10 FOR I=1 TO 8
 20 READ A
 30 BEEP A,.3
 40 NEXT I
 50 RESTORE

 60 GOTO 10
 70 DATA 130,164,195,219,233

 80 DATA 219,195,164
```

Läßt eine Tonleiter erklingen.

Setzt den Datenpointer wieder auf den ersten Wert in der DATA Anweisung mit der kleinsten Nummer (der Wert 130 in Zeile 70).

Die Anweisung RESTORE kann nicht über das Tastenfeld ausgeführt werden.

Das Programm NAMEN

Bisher wurde das Lesen von Daten, die im laufenden Programm eingebettet sind, beschrieben. Das Ihrem HP-75 beiliegende Programm NAMEN benutzt drei wichtige Anweisungen, um auf Informationen in einem Datenfile außerhalb des Programmes zuzugreifen: ASSIGN #, PRINT # und READ #.

Bei Ausführung des Programms NAMEN werden Datenfiles mit Namenslisten und dazugehöriger Information erzeugt. Sie können einen Adressfile oder ein Telefonverzeichnis, eine Liste von Geburtstagen oder Jubiläen oder jede beliebige andere Art von Listen anlegen, wobei bis zu 32 Zeichen Textinformation mit jedem Namen verknüpft werden können.

Suchen Sie die zwei Magnetkarten mit dem Programm NAMEN heraus und führen Sie den Kopierbefehl durch:

```
>copy card to 'namen'
```

```
Copy from card: Align & ERTN
```

Der File NAMEN ist auf Kartenspuren aufgezeichnet – es sind vier Lesedurchgänge erforderlich.

Nachdem der File in den Speicher kopiert ist, erscheinen wieder die Eingabeaufforderung und der Cursor. Führen Sie den Befehl EDIT aus, vorausgesetzt, daß kein nichtleerer Arbeitsfile mehr im Speicher steht (benutzen Sie andernfalls einen der Befehle NAME, RENAME oder PURGE):

```
>edit 'namen'
```

```
NAMEN B 2491 17:19 07/15/82
```

Die Länge und das Datum des Programms können variieren; sie sind von der Version abhängig.

Das vollständig initialisierte Programm benötigt etwa 3000 Bytes des Speichers. Das Programm fragt zu Beginn nach dem Namen, den Sie der Information im Datenfile zuteilen wollen.* Drücken Sie [RUN]:

```
Filename?
                              PRGM
```

Der Filename kann jeder beliebige erlaubte Filename, mit oder ohne Anführungszeichen, sein.

```
Filename? d
                              PRGM
```

Wir verwenden einen einzelnen Buchstaben.

```
Programm  << NAMEN >>
                              PRGM
```

Diese Meldung signalisiert, daß das Programm bereit ist, Information in den File D zu schreiben, dort zu suchen, oder anzuzeigen.

Jede Eingabe kann aus einem Nachnamen, einem Vornamen und bis zu 32 Zeichen Text bestehen. Das Programm NAMEN macht Gebrauch von der Funktion KEY$ (siehe Abschnitt 13) und «übernimmt» das Tastenfeld; die folgenden Tasten erhalten während des Programmablaufs die angegebenen Spezialfunktionen:

[+]	Einfügen eines Eintrags im File.
[↑] und [↓]	Anzeige anderer Einträge in alphabetischer Reihenfolge. Einträge ohne Nachnamen werden vor allen Einträgen mit Nachnamen eingeordnet.
[SHIFT] [↑]	Anzeige des ersten Eintrags des Files.
[SHIFT] [↓]	Anzeige des letzten Eintrags des Files.
[A] - [Z]	Anzeige des ersten Eintrags mit diesem Anfangsbuchstaben des Nachnamens.
[=]	Suche nach einem Eintrag bei gegebenem Vor- und/oder Nachnahmen.
[SHIFT] [=]	Nach [=] benutzt, um den nächsten Eintrag mit dem gleichen Vor- oder Nachnamen anzuzeigen.

* Sie können später den Namen eines existierenden Files eingeben, den Sie auf den neuesten Stand bringen wollen.

[/]	Erzeugt einen «gespaltenen Bildschirm»: die Namen stehen in der linken, die dazugehörige Information in der rechten Bildschirmhälfte. Beim zweiten Drücken wird das Normalformat wieder hergestellt.
[←] und [SHIFT] [←]	Anzeige der linken Schirmhälfte eines Eintrags.
[*]	Ausdrucken des gesamten Files auf den derzeitigen PRINTER IS Einheiten.
[-]	Löschen eines Eintrags.
[.]	Beenden des Programms NAMEN.

Sie können das Programm auch mit [ATTN] unterbrechen. In unserem Beispiel werden wir mit NAMEN ein Telefonverzeichnis anlegen. Drücken Sie [+] und geben Sie den ersten Eintrag in den File D ein – Wilhelm Tell's Namen und Telefonnummer (0041/22/876543, Durchwahl 2341).

```
Nachname :  Tell
Vorname :  Wilhelm
Notiz: 0041/22/876543 x2341


Tell, Wilhelm  0041/22/876543 x2
```

Eingabeaufforderung für den Nachnamen. Geben Sie ihn mit [RTN] ein.

Eingabeaufforderung für den Vornamen.

Eingabeaufforderung für den Text ins Notizfeld. Geben Sie die Telefonnummer in einem bequemen Format ein. Denken Sie daran, daß das Notizfeld maximal 32 Zeichen aufnehmen kann.

Die erste Eingabe ist vollständig. Der Unterschied zwischen Groß- und Kleinschreibung bleibt erhalten.

Holen Sie mit [←] oder [SHIFT] [←] den Rest der Eingabe in die Anzeige:

```
ll, Wilhelm 0041/22/876543 x2341
                        PRGM
```

Die normale, kompakte Anzeigeform für vollständige Einträge.

Erzeugen Sie mit [/] den gespaltenen Schirm:

```
Tell, Wilhelm
                        PRGM
```

Der linke Schirm enthält den Namen.

```
0041/22/876543 x2341
                        PRGM
```

Drücken Sie [←] oder [SHIFT] [←]. Der rechte Schirm enthält die Telefonnummer.

Mit [/] können Sie wieder auf das kompakte Anzeigeformat zurückschalten. Vollständige Einträge bleiben in der Anzeige, bis Sie eine andere Taste drücken.

Im vorhergehenden Beispiel wurden alle Felder ausgefüllt. Sie können Felder übergehen, indem Sie [RTN] drücken, ohne etwas einzugeben. Drücken Sie [+] zur Eingabe des nächsten Eintrags.

```
Nachname :
Vorname :  Gerd
Notiz: x3634
Gerd  x3634
```

Keine Eingabe. Drücken Sie [RTN], um die nächste Eingabeaufforderung zu erhalten.

Eingabe des Vornamens.

Eingabe der Durchwahl.

Vollständiger Eintrag.

Drücken Sie noch einmal [+] für ein drittes Beispiel:

```
Nachname : Tell
Vorname :  Helene
Notiz: 442211
Tell, Helene  442211
```

Eingabe von Helene Tell's Nummer.

Benutzen Sie die Tasten [↑] und [↓], um die Einträge zu überprüfen, solange das Programm läuft. Listen Sie dann die Einträge mit [*]. (Wenn Sie zuvor einen externen Drucker deklariert haben, werden die Einträge ausgedruckt.) Wenn Sie vor [*] die Taste [/] drücken, wird das Notizfeld durch Punkte vom Namensfeld getrennt und von Spalte 33 an gelistet.

Folgende Beschränkungen gelten für die Einträge in NAMEN:

- Die Einträge können jedes Zeichen des Tastenfeldes außer einem Komma enthalten.
- Für jeden Eintrag muß mindestens ein Vor- oder Nachname eingegeben werden.
- Namen werden ohne Rücksicht auf Groß- oder Kleinschreibung sortiert. Eingefügte Leerzeichen können einen Namen jedoch vor die Stelle plazieren, an der er ohne Leerzeichen stehen würde.
- Jedes Feld kann maximal 32 Zeichen enthalten.
- Ein Eintrag kann insgesamt maximal 96 Zeichen umfassen, aber mit [←] kann nur das erste Fenster und mit [→] nur das letzte Fenster angezeigt werden (bei komprimiertem Format).

Wenn Sie einen Eintrag auffinden wollen, müssen Sie entweder die zum ersten Buchstaben des Nachnamens gehörende Taste drücken, oder [=] drücken und den Vor- und/oder Nachnamen eingeben. Wenn keine Übereinstimmung gefunden wird, wird die Programmüberschrift angezeigt. Wenn Sie einen unerwünschten Eintrag entdecken, können Sie ihn mit [−] aus dem Verzeichnis löschen. Beenden Sie schließlich das Programm mit [.] oder [ATTN].

Führen Sie nun den Befehl CAT ALL aus; der File D, der jüngste File, erscheint im Systemkatalog zuerst.

```
D          B    104  11:05  02/15/83
```

Der Katalogeintrag von D, dem Datenfile, der vom Programm NAMEN erzeugt und bearbeitet wurde.

Die Formate, in denen das Programm NAMEN die einzelnen Einträge gespeichert hat, werden beim Listen von D deutlich. Geben Sie `list 'd'` [RTN] ein:

```
1 DATA '',Gerd','x3634'

2 DATA 'Tell','Helene','442211'
3 DATA 'Tell','Wilhelm''0041/2
2/876543 x2341'
```

Ein Nullstring stellt einen Eintrag ohne Nachnamen dar.

Das Programm speichert alle 6543 Einträge – Namen und Zahlen – als Strings in Anführungszeichen.

Starten Sie das Programm wieder mit [RUN], wenn Sie weitere Einträge machen wollen, spezifizieren Sie D als Filenamen, und drücken Sie [+]. Die Anzahl der Einträge im Verzeichnisfile, und auch die Anzahl der Verzeichnisfiles, wird nur durch den verfügbaren Speicherplatz beschränkt.

Erzeugen und Zugriff auf Datenfiles (ASSIGN #)

Die Anweisung ASSIGN # dient zur Erzeugung neuer Datenfiles und zur Vorbereitung existierender Datenfiles auf Ein- und Ausgaben des Programms.

```
ASSIGN # Filenummer TO 'Filename'
```

Die Anweisung ASSIGN # weist dem *Filenamen*, den Sie angeben, die spezifizierte *Filenummer* zu. Die Filenummer kann ein beliebiger numerischer Ausdruck sein, der auf eine ganze Zahl von 1 bis 9999 gerundet wird. Damit kann jedes Programm bis zu 9999 Files adressieren, abhängig vom verfügbaren Speicherplatz. Der Filename kann über einen Stringausdruck spezifiziert werden.

Beispiel:

```
 10 INPUT 'Filename ? '; A$
●20 ASSIGN # 1 to A$
 ⋮
```

Weist dem File A$ die Nummer 1 zu.

Die Anweisung ASSIGN # löst die folgenden Operationen aus:

- Ein neuer BASIC-File dieses Namens wird erzeugt, sofern er nicht schon im Speicher vorhanden ist.
- Der File wird *geöffnet*; das heißt, ein eigener Datenpointer wird auf die erste Zeile des Files oder bei leerem File auf Zeile 0 gesetzt.

Bedeutung der Filenummern:

- Filenummern werden bei der Kommunikation zwischen Programmen und Datenfiles benutzt.
- Es besteht eine Eins-zu-eins Zuordnung zwischen Filenummern und Datenpointern.
- Einem File können nach einer Reihe von ASSIGN # Anweisungen, die den gleichen File spezifizieren, mehrere Datenpointer zugeordnet sein. Sie können mit der Anweisung ASSIGN # auch über das Tastenfeld Filenummern zuweisen.
- Filenummern sind *global*, das heißt, *jedes* Programm kann mit der gleichen Filenummer auf den gleichen File Bezug nehmen.
- Eine Filenummer bleibt einem File unbegrenzt lange zugeteilt, wenn die Nummer nicht in einer anderen ASSIGN # Anweisung einem anderen File zugewiesen wird.

Speichern von Daten in einem File (PRINT #)

Ein mit der ASSIGN # Anweisung geöffneter Datenfile ist für den Zugriff bereit. Die Anweisung PRINT # dient zum Speichern von Werten in einem bestimmten Datenfile.

```
PRINT # Filenummer [, Zeilennummer] ; Ausdruck [, Ausdruck...]
```

Bei jeder Anweisung PRINT # wird der Datenpointer auf den Anfang einer Zeile im spezifizierten Datenfile gesetzt, um den Ausdruck in der PRINT # Liste auszuwerten und die Werte als Datenelemente in den File zu schreiben.

Hier ist ein Beispiel, das Sie über das Tastenfeld ausführen können:

```
assign#2 to 'neu'
```

Drücken Sie RTN; der File NEU wird erzeugt und geöffnet; der Datenpointer wird auf Zeile 0 gesetzt.

```
>print#2;'start',1,sqr(2)
```

Diese Anweisung positioniert den Datenpointer auf Zeile 1 von NEU und schreibt drei Werte in den File.

Führen Sie dann aus:

```
>list 'neu'
```

Listen des Fileeinhalts.

```
1 DATA 'start',1,1.41421356237
```

Zeile 1 des File NEU enthält eine String- und zwei numerische Konstanten.

Wenn in der Anweisung PRINT # keine Zeilennummer spezifiziert wurde, wird der Datenpointer vor dem Schreiben der Daten von seiner jetzigen Position auf die nächste DATA Zeile gesetzt. Wenn nach dem Datenpointer keine DATA Zeile mehr existiert, wird mit dem Befehl PRINT # eine Datenzeile angelegt, deren Zeilennummer um 1 größer als die Nummer der letzten Zeile im File ist. Der File NEU in obenstehendem Beispiel ist bei der Ausführung von PRINT # leer, deshalb erzeugt die Anweisung PRINT # im File die Zeile 1 (um eins größer als Zeile 0).

Ein zweites Beispiel:

```
>print#2;'Zeilenerzeugung'
```

Nach RTN wird im Datenfile Zeile 2 erzeugt, und der Stringinhalt wird in diese Zeile geschrieben.

Falls in der Anweisung PRINT # eine Zeilennummer spezifiziert *ist*, werden die Datenelemente dort abgelegt. Sie können die Zeilennummer mit einem beliebigen numerischen Ausdruck spezifizieren, der auf eine ganze Zahl von 0 bis 9999 gerundet wird. Wenn die Anweisung PRINT # eine nicht existierende Zeile spezifiziert, wird diese zuerst erzeugt und dann beschrieben. Das folgende Beispiel illustriert, wie sich der Datenpointer bewegt, wenn neue Werte in den File NEU geschrieben werden.

```
90 C=2 @ B=200 @ A$='dort'
100 PRINT # 2, 99; 'hier'


110 PRINT # 2; A$
120 PRINT # C, B; 'einmal'
130 PRINT # C; 'mehr'
140 PRINT # C; 3*6,A$
```

Besetzt drei Variablen mit Werten.

Erzeugt Zeile 99 von NEU und speichert dort 'hier'; der Datenpointer bleibt am Fileende stehen.

Erzeugt Zeile 100 und speichert dort 'dort'.

Erzeugt Zeile 200 und speichert dort 'einmal'.

Erzeugt Zeile 201 und speichert dort 'mehr'.

Speichert 18 und 'dort' in Zeile 202.

Nach der Ausführung dieses Programms zeigt ein Listing von NEU:

```
1 DATA 'start',1,1.4121356237
2 DATA 'Zeilenerzeugung'
99 DATA 'hier'
100 DATA 'dort'
200 DATA 'einmal'
201 DATA 'mehr'
202 DATA 18,'dort'
```

Die Datenwerte werden in die spezifizierten Zeilen im File NEU geschrieben.

Die Anweisung PRINT # stellt den Datenpointer zu Beginn auf den Anfang einer DATA Anweisung; existierende Filezeilen können überschrieben werden. Nach einer PRINT # Anweisung bleibt der Datenpointer am Zeilenende stehen.

Falls in der PRINT # Anweisung eine existierende Zeile im File spezifiziert wird, muß diese die Anweisung DATA enthalten; andernfalls erhalten Sie Fehler 34 – no data, es werden keine Daten in diese Zeile geschrieben, und der Datenpointer bleibt am Anfang der spezifizierten Zeile stehen.

Wenn in der PRINT # Liste eine nicht definierte numerische oder Stringvariable erscheint, erhalten Sie die Warnung 7 – no value, und eine Null (bei numerischen Variablen) bzw. ein Nullstring (bei Stringvariablen) wird in den File geschrieben.

Wenn der HP-75 auf DEFAULT OFF gesetzt worden ist, stoppt die Ausführung, ohne daß ein Ersatzwert in den File geschrieben wird.

Wenn die letzte Zeile eines Files die Nummer 9999 besitzt, und eine Anweisung PRINT # versucht, eine Zeile höherer Nummer zu erzeugen, entsteht Fehler 90 – bad line number; es werden keine Daten in den File geschrieben, und der Datenpointer bleibt am Ende von Zeile 9999 stehen.

Wenn Sie versuchen, mit PRINT # den File, der die Anweisung enthält, selbst zu beschreiben, entsteht Fehler 51 – PRINT # to runfile; es werden keine Daten abgespeichert, und das Programm wird deallokatisiert.

Lesen von Daten aus einem File (READ #)

Mit der Anweisung READ # können Sie beliebige Daten in den Speicher einlesen.

```
READ # Filenummer [, Zeilennummer ]; Variable [, Variable...]
```

Die *Filenummer* und die *Zeilennummer* können beliebige numerische Ausdrücke sein, die nur auf ganze Zahlen von 1 bis 9999 bzw. von 0 bis 9999 zu runden sein müssen. Die Variablen können einfache numerische Variablen, numerische Feldelemente, Stringvariablen und Teilstrings sein. Der Datenpointer bewegt sich von links nach rechts über die DATA Anweisung; dabei werden die Variablen in der READ Anweisung von links nach rechts mit den gelesenen Werten besetzt.

Hier sind verschiedene Beispiele, die auf den vorhergehenden PRINT # Anweisungen beruhen:

```
900 READ # 2, 200; U$

910 READ # 2; T$[3,6]

920 READ # 2; V9,T$[7,9]

930 READ # 2, 202; C
940 READ # 2, 99; X$
950 DISP V$;T$[3,6];V9;T$[7,9];C
```

Stellt den Datenpointer auf Zeile 200 und weist U$ den Wert einmal zu.

Weist einem Teilstring von T$ die vier Zeichen mehr des nächsten Datenelements in Zeile 201 zu.

Weist V9 den Wert 18, und einem Teilstring von T$ den Wert dor zu (beide aus Zeile 202).

Weist C ebenfalls Wert 18 aus Zeile 202 zu.

Weist X$ den Wert hier aus Zeile 99 zu.

Zeigt die Variablen an.

Wenn bei einer READ Anweisung der Datenpointer über das letzte Datenelement eines Files hinaus bewegt wird, entsteht Fehler 34 – no data; und der Datenpointer bleibt bei der letzten Datenzeile im File stehen. Ist eine Zeilennummer spezifiziert, und die Variablenliste zwingt den Datenpointer, über das Ende der Zeile hinaus zu springen, entsteht ebenfalls eine Fehlerbedingung, und der Datenpointer bleibt am Zeilenende stehen.

Die Anweisung READ # kann wie auch die Anweisungen ASSIGN # und PRINT # über das Tastenfeld durchgeführt werden. Beachten Sie, daß die in PRINT # und READ # spezifizierten Zeilennummern bei einem RENUMBER Befehl *nicht* geändert werden.

Schließen von Datenfiles

Wenn Sie einen Datenfile nicht mehr benötigen, können Sie ihn mit der Anweisung ASSIGN # schließen.

```
ASSIGN # Filenummer TO *
ASSIGN # Filenummer TO ''
ASSIGN # Filenummer TO '*'
```

Das Schließen eines Datenfiles bewirkt die Aufhebung des Zusammenhangs zwischen Filenummer und Filename. Es ist nicht erforderlich, Datenfiles zu schließen; es gibt jedoch verschiedene Gründe dafür:

- Schutz der Daten in einem File vor überschreiben.
- Einsparung von Speicherplatz. Jeder offene File benötigt 15 Bytes Speicher. Beim Schließen des Files steht dieser Speicher dem HP-75 wieder zur Verfügung.
- Zur Dokumentation in Ihrem Programm, wenn eine Fileoperation abgeschlossen ist.

Sie können nicht mehr benötigte Datenfiles auch löschen, um Speicher zu sparen.

Serieller und direkter Zugriff

Die Spezifikation von Zeilennummern in READ # und PRINT # Anweisungen ist optional und hängt davon ab, ob Sie Ihre DATA Anweisungen einzeln – *direkter Zugriff* – oder sequentiell – *serieller Zugriff* – adressieren wollen. Beim *direkten Zugriff* spezifizieren Sie eine Zeilennummer, beim *seriellen Zugriff* nicht. Es ist oft üblich, den Datenpointer auf eine Anfangszeile im Datenfile zu positionieren, indem Sie die Zeilennummer spezifizieren (direkter Zugriff), um von dort aus seriell zu lesen oder zu schreiben.

Das folgende «Stoppuhr»-Programm ist ein Beispiel für eine serielle Schreiboperation. Das Programm speichert jedesmal, wenn Sie eine Taste drücken, die Anzahl der inzwischen verstrichenen Sekunden in einen Datenfile; Sie können damit soviele Zwischenzeiten nehmen, wie Sie wollen.

```
10 ASSIGN # 3 TO 'UHR'
20 T0=TIME @ I=1
30 IF KEY$='' THEN GOTO 30
40 PRINT # 3 ; TIME-T0

50 DISP #I @ I=I+1
60 GOTO 30
```

Erzeugt und öffnet den Datenfile.

Initialisiert T0 und I.

Das Programm wartet auf einen Tastendruck.

Speichert (seriell) die seit dem Programmstart verstrichene Zeit.

Zeigt und inkrementiert den Zähler.

Programmerklärung. Beim Drücken von [RUN] speichert das Programm die Anfangszeit in der Variablen T0. Von da an springt die Ausführung bei jedem Tastendruck zur Anweisung PRINT # und schreibt die inzwischen verstrichene Zeit in den File UHR. Mit list 'uhr' [RTN] können Sie die Werte im Datenfile untersuchen.

Das folgende Beispiel zeigt, wie Sie mit der Anweisung READ # auf den File UHR direkt zugreifen können.

```
•110 ASSIGN # 45 TO 'uhr'

 120 INPUT 'Zwischenzeit? ';A
•130 READ # 45,A ; B

 140 DISP B;'Sekunden.'
 150 GOTO 120
```

Stellt den Datenpointer 45 auf die erste Zeile im File UHR.

Positionieren Sie den Datenpointer durch Eingabe eines Werts für A; die Daten in dieser Zeile werden eingelesen.

Bei der ersten Ausührung einer seriellen PRINT # Anweisung, oder wenn eine serielle Schreiboperation die Zeilenlänge der momentanen DATA Anweisung überschreitet, reagiert der HP-75 wie folgt:

- entweder wird der Datenpointer auf die Anweisung DATA mit der nächsthöheren Zeilennummer im File gesetzt; dabei werden die Zeilen zwischen den Anweisungen DATA ausgelassen, oder
- es wird eine neue DATA Anweisung im File erzeugt, indem die letzte Zeilennummer im File um 1 erhöht wird. Dies erfolgt, wenn hinter der momentanen Position des Datenpointers keine weiteren DATA Anweisungen mehr vorhanden sind.

Bei seriellen READ # Anweisungen wird der Datenpointer von einem Datenelement zum nächsten bewegt; die Zeilen zwischen den DATA Anweisungen werden ausgelassen.

Mit seriellen PRINT # und READ # Anweisungen können Sie ohne Angabe von Zeilennummern in Datenfiles lesen und schreiben.

Andererseits ermöglichen Ihnen *direkte* PRINT # und READ # Anweisungen, in bestimmten Zeilen zu lesen und zu schreiben. Der Datenpointer kann jedoch nicht über das Zeilenende hinaus bewegt werden. Zuerst wird die in direkten PRINT # oder READ # Anweisungen gegebene Zeile überprüft; wenn die Zeile nicht existiert (nur bei READ #) oder keine DATA Anweisung enthält, erhalten Sie die Fehlermeldung 34 – no data, und der Datenpointer bleibt am Anfang der Zeile stehen. Wenn die Zeilennummer zulässig ist, und die Datenelemente in einer direkten PRINT # Anweisung die Zeilenlänge überschreiten, werden so viele vollständige Datenelemente wie möglich in die Zeile geschrieben, danach entsteht die Fehlerbedingung 28 – record overflow – und der Datenpointer bleibt am Zeilenende stehen. Wenn eine direkte READ # Anweisung versucht, mehr Werte zu lesen, als in der spezifizierten Datenzeile enthalten sind, entsteht Fehler 34 – no data – und der Filepointer bleibt am Zeilenende stehen. Die Beschränkungen bei direktem READ # und PRINT # sollen sicherstellen, daß der Datenpointer nicht weiter wandert als erwartet.

Beachten Sie, daß Sie einen Datenpointer nicht rückwärts in einer DATA Anweisung bewegen können. Jede PRINT # Anweisung schreibt vom Anfang der DATA Anweisung an vorwärts, und jede READ # Anweisung liest von der Stelle des Datenpointers aus vorwärts.

Spezialformen von PRINT # und READ

Es gibt zwei Spezialformen der direkten Anweisungen PRINT # und READ #, die weder Semikolon noch Variablenliste enthalten:

```
PRINT # Filenummer , Zeilennummer
```

```
READ # Filenummer , Zeilennummer
```

Bei der direkten PRINT # Anweisung springt der Datenpointer auf die spezifizierte Zeile und *löscht* sie, bleibt aber auf dieser Zeile stehen. Die spezifizierte Zeile muß entweder eine DATA Anweisung oder eine nicht existierende Zeile sein, andernfalls tritt Fehler 34 – no data – auf.

Beispiel:

```
400 PRINT # 6, 200
```

Stellt den Datenpointer 6 auf Zeile 200, löscht diese Zeile aus dem File und läßt den Datenpointer am Anfang von Zeile 200 stehen.

Die direkte READ # Anweisung bewegt den Datenpointer lediglich auf den Anfang der spezifizierten DATA Anweisung.

Beispiel:

```
410 READ # 6, 1000
```

Positioniert den Datenpointer 6 auf den Anfang der Anweisung DATA in Zeile 1000.

Wenn eine der beiden Anweisungen PRINT # oder READ # eine Zeile ohne DATA Anweisung spezifiziert, oder wenn READ # eine nicht existierende Zeile spezifiziert, entsteht Fehler 34 – no data.

Positionieren des Datenpointers (RESTORE #)

Die Anweisung RESTORE # positioniert den Datenpointer entweder auf eine spezifizierte Zeile eines Files oder auf die DATA Anweisung mit der niedrigsten Zeilennummer im File (falls keine Zeilennummer spezifiziert wurde).

```
RESTORE # Filenummer [, Zeilennummer ]
```

Beide Parameter können numerische Ausdrücke sein. Die *Zeilennummer* muß eine derzeit existierende DATA Anweisung im File adressieren, andernfalls erhalten Sie Fehler 34 – no data.

Die folgenden Anweisungen sind gleichwertig:

```
410 RESTORE # 6, 900
```

```
410 READ # 6, 900
```

Beide Anweisungen positionieren den Datenpointer auf die DATA Anweisung in Zeile 900.

Beachten Sie, daß ein Programm mit den Anweisungen READ # und RESTORE # auf sich selbst operieren kann, genau wie bei den Anweisungen READ und RESTORE.

Zusammenfassung der Anweisungen

Im folgenden finden Sie eine Zusammenfassung der Anweisungen ASSIGN #, PRINT #, READ # und RESTORE #. Die Abkürzungen *fn* und *zn* stehen für Filenummer bzw. Zeilennummer, wobei $1 \leqslant fn \leqslant 9999$ und $0 \leqslant zn \leqslant 9999$ gelten muß.

Anweisung	Wirkung	Fehlerbedingungen
ASSIGN # *fn* TO '*Name*'	Setzt den Datenpointer auf die erste Zeile des spezifizierten Files (Zeile 0 bei einem neu erzeugten File).	Falls Name ein ungültiger Filename oder *fn* nicht im Bereich 1...9999 liegt.
ASSIGN # *fn* TO *	Beendet die Zuordnung zwischen File und Datenpointer.	Falls *fn* nicht im Bereich 1...9999 liegt.
PRINT # *fn*; *Ausgabeliste*	Sucht die nächste DATA Zeile; erzeugt ggf. eine neue DATA Zeile am Fileende. Speichert die Datenelemente und bewegt dabei den Datenpointer durch den File.	Beim Versuch, eine Zeile >9999 zu erzeugen.
PRINT # *fn*, *zn* ; *Ausgabeliste*	Sucht die DATA Zeile *zn*; erzeugt sie ggf. Speichert die Datenelemente und läßt den Datenpointer am Zeilenende stehen.	Falls *zn* existiert, aber keine DATA Zeile ist; oder falls die Elemente die Zeilenlänge überschreiten.
PRINT # *fn*, *zn*	Sucht die DATA Zeile *zn*, löscht sie und stellt den Filepointer an den Zeilenanfang.	Falls *zn* existiert, aber keine DATA Zeile ist.
READ # *fn* ; *Variablenliste*	Liest die Datenelemente ab der Stellung des Datenpointers und bewegt den Datenpointer dabei entlang der Zeile.	Beim Versuch, über das Fileende hinaus zu lesen.
READ # *fn*, *zn* ; *Variablenliste*	Sucht Zeile *zn*. Liest die Datenelemente und läßt den Pointer am Zeilenende stehen.	Bei nichtexistierender Zeile *zn*; falls *zn* keine DATA Zeile ist, oder beim Versuch, über das Zeilenende hinaus zu lesen.
READ # *fn*, *zn*	Stellt den Datenpointer auf den Anfang von *zn*.	Falls *zn* nicht existiert oder keine DATA Zeile ist.
RESTORE # *fn*	Stellt den Filepointer auf den Anfang der ersten DATA Zeile im File.	Falls keine DATA Zeile im File existiert.
RESTORE # *fn*, *zn*	Stellt den Datenpointer auf den Anfang von *zn*.	Falls *zn* nicht existiert oder keine DATA Zeile ist.

Alle PRINT # Anweisungen, RESTORE, RESTORE # und direkten READ # Anweisungen stellen den Datenpointer zu Beginn auf den Anfang einer Zeile.

Beachten Sie, daß die Filenummern in den Anweisungen PRINT #, READ # und RESTORE # sich auf Files beziehen müssen, die zuvor mit ASSIGN # geöffnet worden sind; andernfalls entsteht der Fehler 45 – missing ASSIGN #.

BASIC-Files können sich in den Anweisungen READ # und RESTORE # auf sich selbst beziehen; wenn aber ein Programm versucht, eine Programmoperation auf sich selbst auszuführen, entsteht der Fehler 51 – PRINT # to runfile.

Das Schreiben oder Lesen in privaten BASIC-Files, LEX-Files und Austauschfiles ist nicht gestattet. Wenn ein PB, L oder I File in einer PRINT # oder READ # Anweisung adressiert wird, erhalten Sie Fehler 65 – access restricted. Die in PRINT #, READ # und RESTORE # spezifizierten Zeilennummern werden bei RENUMBER Befehlen *nicht* geändert.

Speichern und Zurücklesen von Feldern

Sie können mit einer einzigen PRINT # bzw. READ # Anweisung ganze Felder abspeichern bzw. einlesen. Ein Komma in der Klammer (zum Beispiel B(,)) symbolisiert ein zweidimensionales Feld.

Beispiele:

```
100 PRINT # 1, A()

200 READ # 1, B(,)
```

Schreibt alle Werte von A(), einem eindimensionalen Feld, in den spezifizierten File.

Weist den Elementen des Feldes B(,) Werte aus dem spezifizierten File zu.

Bei der Anweisung PRINT # wird ein Feld als eine Reihe von Datenelementen abgelegt, vom Element mit der niedrigsten bis zum Element der höchsten Nummer entsprechend der Untergrenze und Dimensionierung des Feldes. Es werden soviele Daten des Feldes in eine DATA Anweisung geschrieben, wie die Zeilenlänge erlaubt. Wenn ein Feld leere oder nicht zugewiesene Elemente enthält, erhalten Sie bei der Ausführung der PRINT # Anweisung die Warnung 7 – no value, und das leere Element erhält den Wert Null zugeteilt (nur bei DEFAULT ON). Sie können die Operation PRINT # mit ATTN unterbrechen.

Bei zweidimensionalen Feldern wird ein Element nach dem anderen geschrieben oder gelesen, wobei der letzte Index am schnellsten läuft; das heißt, zweidimensionale Felder werden zeilenweise abgearbeitet. Das Feld B(,) zum Beispiel besitze die Untergrenze 1 und sei mit folgenden Werten belegt:

B(1,1) = 10	B(1,2) = 20	B(1,3) = 30
B(2,1) = 40	B(2,2) = 50	B(2,3) = 60
B(3,1) = 70	B(3,2) = 80	B(3,3) = 90

Wenn das Feld in Zeile 100 eines Datenfiles geschrieben wird, zeigt eine Auslistung des Datenfiles:

```
100 DATA 10,20,30,40,50,60,70,80,
90
```

Die Werte des Feldes werden zeilenweise gespeichert.

Das folgende Programm besetzt die Feldvariable T(,) mit vier Prüfungsnoten und einem Durchschnittswert dreier Schüler, und speichert dann das Feld T(,) im Datenfile TEST ab.

```
 10 !**Noten und Durchschnitt**
•20 OPTION BASE 1 @ DIM T(3,5)
 30 FOR I=1 TO 3
•40 T(1,5)=0
 50 FOR J=1 TO 4
•60 INPUT T(I,J)
•70 T(I,5)=T(I,5)+T(I,J)/4
 80 NEXT J
 90 NEXT I
•100 ASSIGN # 1 TO 'Test'
•110 PRINT # 1,2 ; T(,)
 120 DISP 'Feld gespeichert'
•130 CAT 'Test' @ LIST 'Test'
```

Dimensioniert T(,).

Initialisiert den Durchschnitt auf Null.

Besetzt die Elemente nacheinander.
Berechnet den neuen Durchschnitt.

Legt den Datenfile TEST an und öffnet ihn.
Speichert das gesamte Feld in Zeile 2 von TEST.

Zeigt den Katalogeintrag an und listet den Datenfile aus.

Führen Sie das Programm aus und listen Sie dann den Datenfile aus – die 12 Prüfungsergebnisse erscheinen in der gleichen Reihenfolge, wie sie über das Tastenfeld eingegeben worden sind; und als fünftes Datenelement erscheint jeweils der Prüfungsdurchschnitt.

Sie können Feldinformation auf die gleiche Weise aus einem Datenfile auslesen. Zum Beispiel:

```
•200 DIM S(3,5) ! OPTION BASE 1
 noch immer wirksam.
•210 READ # 1,2 ; S(,)
 220 DISP 'Durchschnitt';
 230 FOR L=1 TO 3
•240 DISP S(L,5);
 250 NEXT L
 260 DISP
•270 ASSIGN # 1 TO *
```

Deklariert ein weiteres Feld der gleichen Größe und Art wie T(,).
Belegt das gesamte Feld mit den Werten im Datenfile.

Zeigt die Durchschnitte an.

Schließt den File.

Beachten Sie, daß eine Feldvariable in einer READ # Anweisung zeilenweise mit so vielen Datenwerten belegt wird, wie sie Elemente besitzt.

Sie können Felddaten auch elementweise ausgeben oder einlesen.

Beispiele:

```
•300 PRINT # 40 ; B(1,P1),B(2,P2)
 ,B(3,P3)
 ⋮
•500 FOR K=1 TO 10
 510 READ # 30 ; A(K)
 520 NEXT K
```

Speichert drei Werte von B(,).

Weist den Elementen 1 bis 10 von A() Werte aus dem File 30 zu.

Bei großen Feldern werden *serielle* anstelle von *direkten* PRINT # und READ # Anweisungen empfohlen, da serielles Schreiben und Lesen das Lesen und Schreiben von aufeinanderfolgenden DATA Anweisungen erlaubt, während direkte Lese- und Schreiboperationen Fehler aufgrund von Zeilenüberlauf erzeugen können.

Zugriff auf Textfiles

Programme können mit den Anweisungen ASSIGN #, PRINT #, READ # und RESTORE # auch auf *Textfiles* zugreifen und damit Textzeilen lesen, verarbeiten und speichern.

Beispiel:

```
•10 ASSIGN # 1 TO 'Beispiel',TEXT
•20 PRINT # 1; 'Sie koennen
 Literale, Stringvariablen
  und Ausdruecke speichern.'
 30 B$=''
 40 FOR I=97 TO 122
•50 B$=B$&CHR$(I)
 60 NEXT I
 70 FOR J=1 TO 3
•80 PRINT # 1 ; B$

 90 NEXT J
```

Erzeugt und öffnet einen Textfile.
Schreibt die Zeichen eines Literals in den Textfile.

Sammelt die Buchstaben des Alphabets, einen nach dem anderen.

Schreibt die Zeichen der Stringvariablen in den Textfile.

```
1 Sie koennen literale Stringva
riablen und Ausdruecke
speichern.
2abcdefghijklmnopqrstuvwxyz
3abcdefghijklmnopqrstuvwxyz
4abcdefghijklmnopqrstuvwxyz
```

Der Inhalt von B$ wird dreimal gespeichert. Beachten Sie, daß führende Leerzeichen nicht gespeichert werden.

Benutzen Sie die erweiterte Form der ASSIGN # Anweisung, wenn Sie einen Textfile erzeugen und öffnen wollen:

```
ASSIGN # Filenummer TO 'Filename', TEXT
                                   BASIC
```

Die Anweisung ASSIGN # erzeugt einen File des spezifizierten Typs, falls dieser nicht schon existiert, und stellt den Datenpointer auf Zeile 0. Wenn der File schon existiert, braucht TEXT oder BASIC nicht spezifiziert zu werden; die Anweisung ASSIGN # stellt einen Datenpointer auf die erste Zeile des Text- oder BASIC- Files.

Nach dem Öffnen eines Textfiles kann der Fileinhalt mit den Anweisungen PRINT #, READ # und RESTORE # bearbeitet werden:

- mit PRINT # werden die Werte von Stringausdrücken in den File geschrieben.
- mit READ # werden Zeichen aus dem File in Stringvariablen und Teilstrings eingelesen.
- mit RESTORE # wird der Datenpointer auf die erste Zeile des Files oder auf den Anfang der spezifizierten Zeile gesetzt.

Hier sind die wesentlichen Unterschiede zwischen dem Zugriff auf BASIC und auf Textfiles:

Unterschied	Geöffneter BASIC-File	Geöffneter Textfile
Zusammensetzung	Besteht aus 0 oder mehr numerierten Zeilen mit DATA Anweisungen, die unter andere Programmanweisungen gemischt sein können.	Besteht aus 0 oder mehr numerierten Zeilen mit Text.
Datenelemente	Können Zeichen oder Zahlen sein, durch Komma getrennt.	Jede Zeile wird als ein Datenelement behandelt, d.h. als fortlaufender Zeichenstring.
Fehlerbedingungen	Falls die in PRINT #, READ # oder RESTORE # spezifizierte Zeile existiert, aber keine DATA Anweisung enthält.	Falls die in READ # oder RESTORE # spezifizierte Zeile nicht existiert.

Eine an einen Textfile adressierte Liste von PRINT # Daten darf nur aus Strings bestehen; es sind Literale, Stringvariablen, Teilstrings, Stringfunktionen oder eine beliebige Kombination daraus zulässig. Ein Versuch, einen numerischen Wert in einen Textfile zu schreiben, erzeugt Fehler 65 – access restricted. Wenn zwei oder mehrere Strings in der gleichen PRINT # Anweisung erscheinen, werden die Zeichen beider Strings in der gleichen Zeile des Textfiles gespeichert. Wenn Sie mit PRINT # Textfiles von BASIC-Programmen aus beschreiben, ist es eine gute Angewohnheit, als erstes Zeichen jeder Textzeile ein Leerzeichen zu setzen. Wenn Ziffern direkt, ohne trennende Leerstelle, auf eine Zeilennummer folgen, interpretiert das Betriebssystem den gesamten Zahlenstring als Zeilennummer, wenn der Befehl PLIST ausgeführt wird. Solche Zeilen werden entweder gar nicht oder fehlerhaft und in der falschen Reihenfolge ausgelistet.

Im folgenden Beispiel wird angenommen, daß der Textfile NOTIZ existiert, und daß A$ und B$ zuvor deklarierte Stringvariablen sind:

```
100 ASSIGN # 3 TO 'notiz'
120 PRINT # 3,100 ; A$,B$


130 PRINT # 3; A$[1,8]&UPRC$(B$)



140 PRINT # 3, 99

150 RESTORE # 3
```

Öffnet den Textfile.

Die Anweisung PRINT # sucht bzw. erzeugt die Zeile 100 und füllt sie mit den Werten beider Variablen.

Die Anweisung PRINT # (seriell) schreibt den Wert des Stringausdrucks in die nächste Zeile des Files und erzeugt, falls erforderlich, am Fileende eine neue Zeile.

Löscht Zeile 99, läßt den Datenpointer aber am Anfang dieser Zeile stehen.

Stellt den Datenpointer auf die erste Zeile des Files zurück.

Wenn ein Datenelement in der PRINT # Anweisung die Zeilenlänge des Textfiles überschreitet, gilt:

- in einer *seriellen* Schreiboperation wird der Datenpointer auf die nächste Zeile des Files gesetzt (falls erforderlich wird am Fileende eine neue Zeile erzeugt), und der Schreibvorgang wird in der neuen Zeile fortgesetzt.
- in einer *direkten* Schreibeoperation bleibt der Datenpointer am Ende der laufenden Zeile stehen, Sie erhalten die Fehlermeldung 28 – record overflow, und das momentane sowie nachfolgende Datenelemente werden nicht in den File geschrieben.

Jede Stringvariable in einer READ # Liste wird mit den Zeichen einer Textzeile belegt. Enthält die Anweisung READ # mehrere Stringvariablen, dann werden aufeinanderfolgende Textzeilen eingelesen.

Beispiele:

```
300 DIM C$[96],D$[96]
310 READ # 1,100 ; C$
320 READ # 1 ; D$
330 READ # 1 ; C$,D$
```

Dimensioniert die Variablen C$ und D$.

Liest alle Zeichen von Zeile 100 in C$ ein.

Liest alle Zeichen der nächsten Zeile in D$ ein.

Liest alle Zeichen der nächsten Zeile in C$ und der darauffolgenden Zeile in D$ ein.

Nach einer READ # Anweisung auf einem Textfile bleibt der Datenpointer auf dem Zeilenende der gerade gelesenen Textzeile positioniert.

Folgende Fehler können beim Lesen von Textfiles auftreten:

- Fehler 42 – string too long – tritt auf, wenn eine Stringvariable in einer READ # Anweisung nicht groß genug dimensioniert wurde, um alle Zeichen einer vom Textfile eingelesenen Zeile aufzunehmen.
- Die READ # Liste darf nur aus Stringvariablen und Teilstrings bestehen; Fehler 65 – access restricted – tritt auf, wenn versucht wird, Text in eine numerische Variable einzulesen.
- Fehler 34 – no data – tritt auf, wenn die in einer READ # Anweisung spezifizierte optionale Zeilennummer im File nicht existiert.

- In einer direkten Leseoperation darf der Datenpointer nicht über das Zeilenende hinaus bewegt werden. Deshalb dürfen in einer direkten READ # Anweisung keine mehrfachen Variablen spezifiziert werden.

Das Programm SUCH

Das folgende Programm benutzt die Anweisungen ASSIGN #, PRINT #, READ # und RESTORE #, um Stringmuster in einem Textfile mit Namen QUELLE zu suchen.
Nach der Initialisierung belegt das Programm mit Kommentaren etwa 1200 Bytes.

Das Programm SUCH

```
10 ! Programm SUCH
20 DELAY 2 @ WIDTH INF
30 DIM A$[96] ! A$=Zeile; S$=Suc
hstring
40 ASSIGN # 10 TO 'quelle'
50 PRINT # 10,9999 ; 'ENDE' ! Se
tzt Endmarkierung.
60 RESTORE # 10 @ C=0 @ P=0 ! In
itialisiert Pointer, Zeilenzaehl
er und Spaltenposition.
70 INPUT 'Suchstring: ';S$
80 READ # 10 ; A$
90 IF A$='ENDE' THEN 150 ! Filee
nde erreicht.
100 C=C+1 ! Erhoeht den Zaehler.
110 P=POS(UPRC$(A$),UPRC$(S$)) !
 P ist 0, wenn die Zeile S$ nich
tenthaelt, sonsten die Spalte.
120 IF P<>0 THEN E=LEN(S$)+P-1 @
 GOTO 150 ! String gefunden, E i
st die Endposition.
130 READ # 10 ; A$ ! Liest die n
aechste Zeile.
140 GOTO 90 ! Untersucht die gel
esene Zeile.
150 DISP C; ! Anzeige des Zeilen
zaehlers.
160 IF P<>0 THEN A$[P,E]=FNU$(A$
[P,E]) @ DISP ': ';A$ ELSE DISP
'Zeilen - nicht gefunden.'
170 PRINT # 10,9999 ! Loescht di
e Endmarkierung.
180 ASSIGN # 10 TO * ! Schliesst
 den File.
190 END
200 DEF FNU$(T$) ! Unterstreicht
 T$.
210 FOR K=1 TO LEN(S$)
220 T$[K,K]=CHR$(NUM(T$[K])+128)
 ! Unterstreicht einzelnes Zeich
en.
230 NEXT K
240 FNU$=T$
250 END DEF
```

Der Textfile QUELLE

```
10 Dieser File kann aus beliebig
en Zeichenfolgen zusammengesetzt
20 sein, die entweder ueber das
30 Tastenfeld oder ueber ein
40 anderes BASIC Programm abge-
50 speichert werden koennen.
```

Das Programm SUCH schreibt zuerst eine Endmarkierung in Zeile 9999 des Files QUELLE, damit das Programm die Suche beendet, wenn das Fileende erreicht wird. Bei einer Stringeingabe von null oder mehr Zeichen sucht das Programm nach dem ersten Auftreten des Strings im File QUELLE, in Groß- oder Kleinschrift. Wenn das Muster gefunden wurde, zeigt das Programm die Nummer der Zeile, in der es auftritt (relativ zur ersten Zeile des Files QUELLE), und danach die ganze Zeile an, – das Muster wird unterstrichen. Wenn der spezifizierte String im File nicht gefunden wird, zeigt das Programm die Anzahl Zeilen des Files und die Meldung nicht gefunden an.

Bei der Ausführung fragt das Programm nach dem Suchstring:

```
Suchstring:
                    PRGM
```

Suchen Sie nach dem Wort BASIC (angenommen, Ihr File QUELLE hat den gezeigten Inhalt).

```
Suchstring: basic
                    PRGM
```

```
4: anderes BASIC Programm a
```

Der String steht in der vierten Zeile des Textfiles.

Versuchen Sie einen weiteren Programmlauf mit einem nicht vorkommenden Zeichenmuster:

```
Suchstring: Unsinn
                    PRGM
```

```
5 Zeilen - nicht gefunden
```

Lange Datenzeilen

In eine Anzeigezeile können maximal 94 Zeichen eingegeben werden; zwei Positionen sind für die Eingabeaufforderung und den Cursor reserviert. Mit der Anweisung PRINT # können Sie jedoch mehr als doppelt so lange Datenzeilen in BASIC- und Textfiles schreiben.

Die maximale Zeilenlänge in einem BASIC-File beträgt: 253 Bytes. Jedes Datenelement in der DATA Anweisung belegt zwei oder mehr Bytes der verfügbaren 253 Bytes:

- Jede ganze Zahl belegt 4 Bytes.
- Jede Dezimalzahl belegt 9 Bytes.
- Jeder String belegt 2 Bytes plus 1 Byte für jedes Zeichen des Strings.

Beachten Sie, daß keine Bytes für die Zeilennummer, das Schlüsselwort DATA, Leerzeichen, Anführungszeichen oder Kommata benötigt werden.

In einem Textfile beträgt die maximale Zeilenlänge 255 Zeichen einschließlich der Zeilennummer.

Wenn eine lange Datenzeile danach gelistet oder eingelesen wird, erhalten Sie die Warnung 76 – line too long, und nur die ersten 94 Zeichen der Zeile werden angezeigt. Das Ende einer langen Zeile kann nicht direkt untersucht oder editiert werden, mit der Anweisung READ oder READ # kann jedoch auf alle Werte der Zeile zugegriffen werden.

Fehlerbedingungen:

- Wenn bei einer PRINT # Anweisung in direktem Zugriff mehr als 253 Bytes in einen BASIC-File geschrieben werden, tritt Fehler 28 – record overflow – auf, und es werden nur so viele Datenelemente wie möglich in die Zeile geschrieben.
- Wenn bei einer PRINT # Anweisung in direktem Zugriff mehr als 255 Zeichen in einen Textfile geschrieben werden, tritt Fehler 28 – record overflow – auf, und es werden nur so viele Stringausdrücke wie möglich in die Zeile geschrieben.

Im seriellen Zugriff dagegen wird bei der Anweisung PRINT # der Datenpointer auf die nächste verfügbare Zeile gestellt und der Rest der Datenelemente dorthin geschrieben.

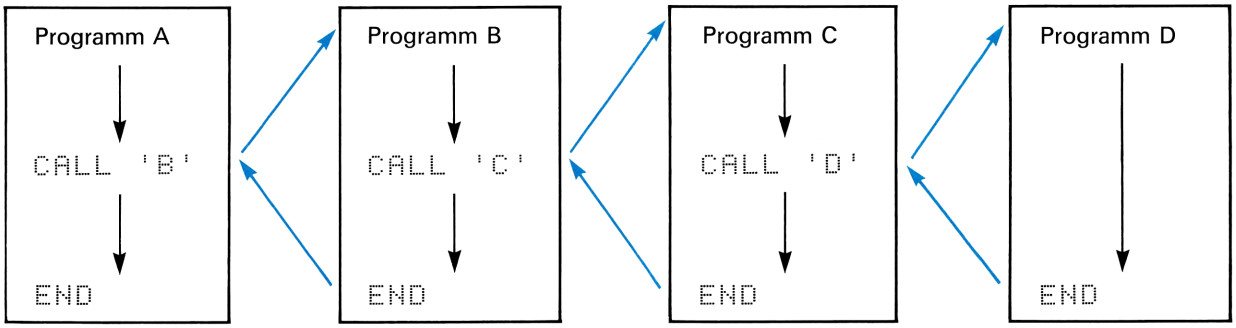
Abschnitt 15

Aufrufen von Programmen

Inhalt

Einführung

Jedes Programm kann ein anderes Programm *aufrufen*, oder ausführen. Dieses Programm wiederum kann ein weiteres Programm aufrufen usw.



Wenn ein aufgerufenes Programm beendet ist, gibt es die Programmkontrolle an das Programm, von dem es aufgerufen worden ist, an die auf CALL folgende Anweisung zurück. Hauptprogramme können deshalb aus einer Reihe von Aufrufen von *Routinen*, oft auch *Unterprogramme* genannt, bestehen. Oft benutzte Unterprogramme können als «Bibliotheksroutinen», von allen Programmen zugreifbar, im Speicher abgelegt werden. Werte können mit Hilfe von Datenfiles zwischen Programmen übermittelt werden; zusätzlich können die Parameter per *Adresse* oder per *Wert* übergeben werden.

Die Anweisungen CALL und END

Programmaufrufe erzeugen eine unbedingte Verzweigung der Programmausführung. Fügen Sie die Anweisung CALL an die Stelle im aufrufenden Programm, von der aus die Verzweigung erfolgen soll.

```
CALL 'Filename'
```

Der *Filename* kann durch einen beliebigen Stringausdruck, der einen existierenden Programmfile im Speicher benennt, spezifiziert werden. Jeder ausführbare File kann auch aufgerufen werden.

Wenn das aufgerufene Programm die Anweisung END oder die End-Of-File Markierung antrifft, gibt es die Ausführung an das aufrufende Programm und zwar an die erste Anweisung nach CALL zurück. Die Anzahl von Programmaufrufen ohne zugehörige Rücksprünge ist nur durch den verfügbaren Speicherplatz beschränkt.

Vergleich von RUN und CALL

Die Anweisungen RUN und CALL haben einige gemeinsame Eigenschaften:

- Weder RUN noch CALL beeinflussen die Werte von Rechnervariablen.
- Sowohl RUN als auch CALL initialisieren die Werte von Programmvariablen.
- Die mit beiden Anweisungen gestarteten Programme werden nach der *Beendigung* der Ausführung deallokatisiert, und die Variablenwerte gehen verloren; beide Programmtypen bleiben initialisiert, wenn die Ausführung mit ATTN oder STOP *unterbrochen* wird.
- Sowohl RUN als auch CALL können über das Tastenfeld ausgeführt werden.
- Jedes beliebige Programm kann mit RUN oder CALL jedes beliebige andere Programm starten.
- Der Filelepointer bleibt auf den momentanen EDIT-File positioniert, wenn RUN oder CALL ausgeführt werden.

Die Unterschiede sind:

- RUN kann eine Anfangszeile spezifizieren; CALL beginnt die Programmausführung immer bei der Zeile mit der niedrigsten Nummer eines Programms.
- Die Ausführung kehrt nicht zu einem Programm zurück, das ein anderes mit RUN aufruft.
- Die Werte der Variablen eines mit CALL aufrufenden Programms bleiben erhalten, während das aufgerufene Programm ausgeführt wird. RUN bedingt eine Deallokation aller Programmvariablen, die nicht zum derzeit laufenden Programm gehören.

Die Programme EINS und ZWEI

Anhand der folgenden kurzen Programme EINS und ZWEI soll eine einfache Aufrufoperation illustriert werden. Beginnen Sie mit ZWEI: Geben Sie `edit 'zwei'` RTN und danach das vierzeilige Programm ein.

```
10 DISP 'Dies ist ZWEI.'
20 A=2
30 B$='zwei'
40 STOP
```

Anzeige einer Meldung und Zuweisung zweier Variablen.

Unterbrechen der Ausführung.

Schreiben Sie jetzt das Programm EINS: Geben Sie `edit 'eins'` RTN und danach das Programm ein.

```
 10 DISP 'Dies ist EINS.'
 20 A=1
 30 B$='eins'
 40 CALL 'zwei'

 50 DISP 'Zurueck in EINS.'
 60 STOP
```

Die Ausführung wird an ZWEI abgegeben und kehrt danach zu Zeile 50 zurück.

Drücken Sie nach der Eingabe beider Programme RUN:

```
Dies ist EINS.
Dies ist ZWEI.
```

Die Ausführung stoppt bei Zeile 40 von ZWEI. Der Filepointer bleibt jedoch in EINS auf der Zeile stehen, wo er sich vor dem Drücken von RTN befunden hat.

Bestimmen Sie die Stellung des Filepointers, indem Sie CAT RTN eingeben, um den Namen des Files zu erhalten, und FETCH RTN, um die laufende Zeile zu erhalten. Drücken Sie dann ATTN und prüfen Sie die Werte der Variablen A und B$:

```
>a,b$
```

```
 2                              zwei
```

Die Werte, die im zweiten Programm zugewiesen worden sind.

Das Beispiel soll zeigen, daß Sie nur die Werte des unterbrochenen Programms (hier: ZWEI) prüfen und ändern können, auch wenn der Filepointer in einem anderen Programm (hier: das aufrufende Programm EINS) steht. Die Werte des aufrufenden Programms bleiben jedoch im Speicher erhalten, bis die Ausführung zum aufrufenden Programm zurückkehrt.

Beachten Sie auch, daß Sie die Werte von derzeitigen *Rechnervariablen* während einer Programmunterbrechung prüfen und ändern können, falls diese nicht die gleichen Namen wie die Variablen des unterbrochenen Programms besitzen.

Nehmen Sie die Programmausführung mit cont RTN wieder auf:

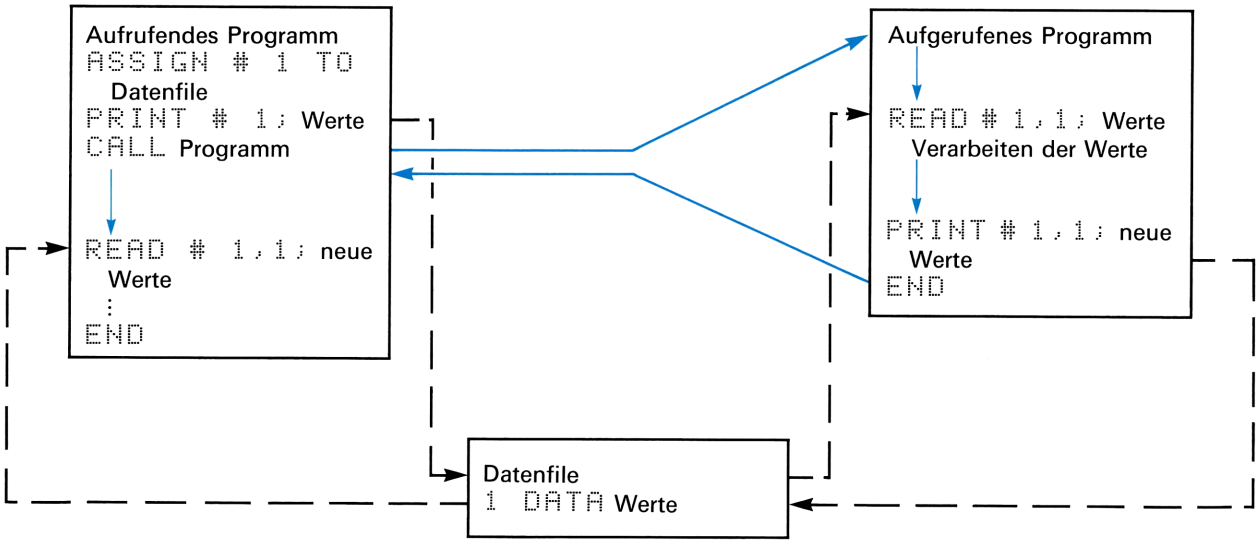
```
Zurueck in EINS.
```

Die Ausführung wird nun bei der Anweisung STOP in Zeile 60 von EINS unterbrochen. Wenn Sie die Werte von A und B$ prüfen, werden Sie feststellen, daß diese ihre ursprünglichen Werte 1 und eins wieder angenommen haben. Bei der Fortsetzung des zweiten Programms wurde sofort die End-Of-Line Markierung gefunden, wodurch die Ausführung an Zeile 50 in EINS zurückgegeben wurde. Danach wurde das Programm ZWEI deallokatisiert, dessen Variablenwerte gingen verloren.

Wenn nach einem Programmaufruf ein Fehler auftritt, stoppt das aufgerufene Programm bei der Zeile, in der der Fehler aufgetreten ist und zeigt eine Fehlermeldung an; der Filepointer bleibt auf dem File, der vor der Programmausführung editiert wurde, positioniert.

Übergabe von Werten zwischen Programmen (ASSIGN #, PRINT #, READ #)

Mit Hilfe von Datenfiles können Programme auf eine unbeschränkte Anzahl von gemeinsamen Werten zugreifen.



Sie sehen in diesem Diagramm, daß das aufrufende Programm einen Datenfile anlegt, in den es die weiterzureichenden Werte einträgt, und dann das zweite Programm aufruft. Das aufgerufene Programm liest die Werte im Datenfile,

verarbeitet sie und schreibt die neuen Werte in den Datenfile. Nachdem die Ausführung wieder an das aufrufende Programm zurückgegeben wurde, liest dieses die neuen Werte vom Datenfile.

Sie können für den Datenfile beliebige Filenamen, Filenummern und Filezeilen spezifizieren, solange das aufrufende und das aufgerufene Programm auf die selben Datenelemente zugreifen. Mit den folgenden zwei Programmen werden die Polarkoordinaten eines Punktes aus seinen kartesischen Koordinaten berechnet. Geben Sie `edit 'haupt'` RTN und dann das aufrufende Programm ein:

```
 10 ! --HAUPTPROGRAMM--
 20 OPTION ANGLE RADIANS
 30 INPUT 'Kartesische Koordinate
 n:';X,Y
•40 ASSIGN # 1 TO 'daten'

•50 PRINT # 1 ; X,Y
 60 CALL 'polar'
•70 READ # 1,2 ; R,T
 80 DISP 'Die Polarkoordinaten vo
 n';X;Y;'sind';R;T
```

`DATEN` speichert kurzzeitig die Werte der Koordinaten.

Schreibt die Werte von `X` und `Y` in den Datenfile.

Liest die neuen Werte vom Datenfile.

Geben Sie dann `edit 'polar'` RTN und die Routine `POLAR` ein:

```
 10 ! --POLAR--
•20 READ # 1,1 ; X9,Y9
 30 R9=SQR(X9^2+Y9^2)
 40 T9=ANGLE(X9,Y9)
•50 PRINT # 1,2 ; R9,T9
 60 END
```

Greift auf die Werte des Hauptprogramms zu.

Schreibt neue Werte in den Datenfile.

Starten Sie nun das Programm `HAUPT`:

```
Kartesische Koordinaten:
                    PRGM
```

Geben Sie `3,4` ein und drücken Sie RTN.

```
Die Polarkoordinaten von 3 4
sind 5    .927295218002
```

Beachten Sie, daß die Routine `POLAR` aus Zeile 1 des Datenfiles einliest und in Zeile 2 schreibt. Es könnte stattdessen auch in der gleichen Zeile lesen und schreiben, falls das Hauptprogramm auch auf diese Zeile zugreift.

Beachten Sie auch, daß die Werte der Parameter `X` und `Y` des Hauptprogramms nicht von der Veränderung dieser Werte im aufgerufenen Programm betroffen sind. Dies wird als Übergabe per *Wert* bezeichnet. Es ist auch die Übergabe per *Adresse* – wenn die Parameterwerte des aufrufenden Programms vom aufgerufenen Programm verändert *werden* – möglich; das aufrufende Programm muß dazu beim Einlesen der neuen Werte vom Datenfile diese verarbeiteten Werte den ursprünglichen Parametern zuweisen.

Globale und lokale Deklarationen

Jedes Programm kann Rechnerzustände wie zum Beispiel die Verzögerungsrate oder den trigonometrischen Modus setzen. Eine *globale* Deklaration (zum Beispiel `OPTION ANGLE RADIANS`) bleibt erhalten, bis eine andere globale Deklaration (zum Beispiel `OPTION ANGLE DEGREES`) durchgeführt wird.

Alle Befehle, die Rechnerzustände setzen (Seite 163) sowie bestimmte BASIC-Anweisungen wirken global. Zum Beispiel sind die in der Anweisung ASSIGN # deklarierten Filenummern global – jeder geöffnete Datenfile kann von jedem Programm wie auch über das Tastenfeld bearbeitet werden. Wenn die Anweisung ASSIGN # einmal durchgeführt ist, bleibt der spezifizierte File solange mit der spezifizierten Filenummer verknüpft, bis die Filenummer in einer weiteren Anweisung ASSIGN # umdeklariert wird.

Andere Anweisungen, sog. *lokale* Deklarationen, wirken nur auf den Ablauf des derzeit ausgeführten Programms. Die Anweisungen DIM, IMAGE, LET und OPTION BASE sind Beispiele für lokale Deklarationen. Verzweigungsanweisungen, Unterprogramme und benutzerdefinierte Funktionen sind nur lokal im momentan ausgeführten Programm definiert.

Programmtimer operieren ebenfalls lokal. Wenn die Ausführung mit der Anweisung CALL übertragen wird, zählt ein aktiver Timer weiter, kann aber nicht mehr unterbrechen. Wenn die Ausführung wieder zum aufrufenden Programm zurückkehrt, unterbricht der Timer dieses nach Ablauf des nächsten Timerintervalls. Wenn ein Timer zum Beispiel auf ein 10-Sekunden-Intervall gesetzt ist, und fünf Sekunden vor Ablauf der Frist ein anderes Programm aufgerufen wird, welches 7 Sekunden benötigt, dann wird der Timer während des Aufrufs ignoriert, und die nächste Unterbrechung erfolgt 8 Sekunden nach der Rückkehr der Programmkontrolle zum aufrufenden Programm.

Timeraufrufe sind zwar lokal, *Timernummern* aber sind global. Es kann zum Beispiel nur einen Timer #33 geben. Wenn ein Programm Timer #33 deklariert und dann ein anderes Programm aufruft, und dieses ebenfalls Timer #33 deklariert, dann wird dieser Timer das «Eigentum» des aufgerufenen Programms. Das aufrufende Programm muß danach den Timer wieder deklarieren, um ihn wieder kontrollieren zu können. Das bedeutet, daß ein aufgerufenes Programm einen in einem anderen Programm gesetzten Timer desaktivieren kann.

Die von der Funktion RND erzeugte Folge von Zufallszahlen (Seite 83) wird durch Programmaufrufe mit CALL nicht unterbrochen. Wenn zum Beispiel Programm A drei Zufallszahlen x_1, x_2 und x_3 anzeigt und Programm B aufruft, ist die nächste für B verfügbare Zufallszahl die vierte Zahl in der Folge, x_4 (falls B nicht die Anweisung RANDOMIZE enthält, die eine neue Folge startet). Wenn B nicht aufgerufen, sondern gestartet wird (RUN), beginnt die Folge von Zufallszahlen mit einem in B lokalen Startwert.

Schlagen Sie die Beschreibung der Deklarationen ON ERROR, OFF ERROR, TRACE FLOW, TRACE VARS und TRACE OFF in Abschnitt 17 nach.

Rekursive Aufrufe

Die Anweisung CALL erlaubt *rekursive Aufrufe* – Routinen können *sich selbst* aufrufen.

Beispiel: Die Fakultätsfunktion kann folgendermaßen rekursiv definiert werden:

Für alle nicht negativen ganzen Zahlen x gilt

$$\text{FAK}(x) = \begin{cases} 1 & \text{falls } x = 0. \\ x \times \text{FAK}(x-1) & \text{sonst.} \end{cases}$$

Diese Definition ist rekursiv, da im zweiten Fall die Funktion durch sich selbst definiert ist.

In den folgenden zwei Programmen sehen Sie, wie mit rekursiven Aufrufen Fakultäten berechnet werden können. Geben Sie edit 'hprog' RTN und das Hauptprogramm ein:

```
 10 ! --HPROG--
•20 ASSIGN # 1 TO 'temp'
 30 INPUT 'Fakultaet von? ';M
•40 M=ABS(IP(M))
•50 PRINT # 1,1 ; M
```

Öffnet einen Datenfile, um Werte weiterzugeben.

Wandelt M in eine positive ganze Zahl um.
Schreibt den Wert M in den Datenfile.

```
 60 CALL 'fak'
•70 READ # 1,1 ; N
 80 DISP M;'Fakultaet ist';N
 90 GOTO 30
```

Liest den aus der letzten Ausführung von FAK resultierenden Wert.

Geben Sie dann mit edit 'fak' RTN die rekursive Routine ein:

```
10 ! --fak--
20 READ # 1,1 ; X
30 IF X=0 THEN PRINT # 1,1 ; 1 @
 END
40 PRINT # 1,1 ; X-1

50 CALL 'fak'
60 READ # 1,1 ; Y



70 X=X*Y
80 PRINT # 1,1 ; X

90 END
```

X erhält seinen Wert vom aufrufenden Programm.

Prüft den Parameter X auf die Endbedingung.

Andernfalls wird X dekrementiert und der neue Wert in den File geschrieben.

Der Rekursivaufruf.

Nach dem Aufruf kehrt die Ausführung hierher zurück. Y erhält den Wert der vorhergehenden Routine.

Berechnet die Fakultät für die derzeitige Rekursionsebene.

Speichert den momentanen Wert der Fakultät in den Datenfile zurück.

Gibt die Ausführung an das aufrufende Programm zurück.

Starten Sie HPROG, nachdem Sie die beiden Programme eingegeben haben, und berechnen Sie die Fakultät von 6:

```
Fakultät von? ▮
                    PRGM
```

Geben Sie 6 ein und drücken Sie RTN.

```
6 Fakultaet ist 720
```

Die Antwort, nach 6 Aufrufen der Routine FAK.

Sie können untersuchen, wie die Größe des verfügbaren Speichers mit aufeinanderfolgenden n verändert wird. Fügen Sie in der Routine FAK eine Zeile 15 ein:

```
15 DISP MEM
```

Zeigt die Größe des verfügbaren Speichers direkt nach dem Aufruf der Routine an.

Wenn Sie HPROG jetzt starten, werden Sie nach jedem Aufruf eine Abnahme des verfügbaren Speichers feststellen; der Speicher wurde mit Werten von Programmvariablen, Rücksprungadressen usw. gefüllt. Die Anzahl ineinander verschachtelter Programmaufrufe ist nur durch den verfügbaren Speicherplatz beschränkt.

Es sind noch andere Rekursionsformen möglich:

Gegenseitige Rekursion

A ruft B auf
B ruft A auf

Indirekte Rekursion

A ruft B auf
B ruft C auf
C ruft A auf

Ein einfaches Beispiel für gegenseitigen Aufruf:

Das Programm ALFRED.

```
10 DISP 'Alfred ruft Maria'
20 CALL 'Maria'
```

Das Programm MARIA.

```
10 DISP 'Maria ruft Alfred'
20 CALL 'Alfred'
```

Diese zwei Programme erzeugen eine endlose Folge von Programmaufrufen, bis die Ausführung unterbrochen wird (mit ATTN, oder bei gefülltem Speicher: Fehler 16 – not enough memory). Beachten Sie, daß beim Editieren eines der beiden Programme *beide* deallokatisiert werden; der für Variablen und Rückkehradressen benutzte Speicher wird wieder an das System zurückgegeben. (Die Programme werden auch deallokatisiert, wenn Sie ein anderes Programm starten oder clear vars RTN eingeben.)

Mit der Funktion CAT$, die in Abschnitt 13 besprochen wurde, können Sie den Programmfluß nachvollziehen, wenn die Programmkontrolle zwischen Programmen hin- und hergegeben wird. Erinnern Sie sich, daß CAT$ mit negativem Argument den Katalogeintrag des derzeit laufenden Programms zurückgibt. Wenn Sie CAT$(-1) an den Anfang einer Routine stellen, erhalten Sie den Katalogeintrag der Routine angezeigt, sobald deren Ausführung beginnt.

Abschnitt 16

Anzeige- und Druckformatierung

Inhalt

Einführung

Sie haben zur Kontrolle von Anzeige- und Druckausgaben zur Verfügung:

- Kommata, Semikolons, und die Funktion TAB in den Anweisungen DISP und PRINT.
- die Befehle WIDTH und PWIDTH.
- Kontrollzeichen und Escapecodes.
- Die Anweisung ENDLINE (nur bei PRINT und PLIST).

Die drei zusätzlichen Anweisungen DISP USING, PRINT USING und IMAGE geben Ihnen noch weitere Möglichkeiten, die Ausgaben auf DISPLAY IS und PRINTER IS Einheiten zu formatieren.

Verwendung von IMAGE

Die Anweisung IMAGE spezifiziert das Format, in welchem Zahlen und Strings in den Anweisungen DISP USING und PRINT USING angezeigt oder ausgedruckt werden.

```
DISP USING Zeilennummer [, Ausgabeliste ]
```

```
PRINT USING Zeilennummer [, Ausgabeliste ]
```

```
IMAGE Formatstring
```

Die *Zeilennummer* in der DISP USING oder PRINT USING Anweisung muß eine vorzeichenlose ganze Zahl von 0 bis 9999 sein und sich auf eine gültige IMAGE Anweisung beziehen, die irgendwo im Programm stehen kann. Die DISP USING Liste und die PRINT USING Liste können aus beliebigen numerischen oder Stringausdrücken zusammengesetzt sein.

Der *Formatstring* in der IMAGE Anweisung besteht aus einer Reihe von Zeichen, die das oder die gewünschten Ausgabeformate spezifizieren. Der HP-75 akzeptiert nach dem Schlüsselwort IMAGE *jede* Kombination von Zeichen mit oder ohne Anführungszeichen. Der HP-75 prüft die Anweisung IMAGE bei deren Ausführung in einem Programm, und nicht bei deren Eingabe, auf korrekte Spezifikation der Formate. Die Anweisung IMAGE sollte nicht in einer Zeile mit Mehrfachanweisungen stehen; wenn sie einer anderen Anweisung folgt, wird sie nie ausgeführt, und wenn ihr eine weitere Anweisung folgt, wird diese nicht als Anweisung erkannt. Eine über das Tastenfeld ausgeführte IMAGE Anweisung wird ignoriert.

Die Datenelemente in der PRINT USING oder DISP USING Liste sind durch Kommata oder Semikolons zu trennen. Diese beeinflussen jedoch nicht das Format wie in den Anweisungen DISP und PRINT; sie trennen nur die Elemente der Liste. Die Ausgabe wird durch den *Formatstring* in der IMAGE Anweisung vollständig kontrolliert.

Der IMAGE Formatstring (der *Bild-Formatstring*) ist aus *Feldspezifikatoren* zusammengesetzt, die durch Begrenzer getrennt sind. Jeder Spezifikator ist aus Symbolen zusammengesetzt, die das Format eines einzelnen Datenelements der DISP USING oder PRINT USING Liste bestimmen. Die Symbole in der IMAGE Anweisung spezifizieren die Anzahl von Stellen, die Stellung eines Kommas, Dezimalpunkts oder Leerzeichens – praktisch alles, was mit der Ausgabe von Strings und Zahlen zu tun hat.

Beispiel:

```
300 IMAGE dddcddd.dd, 10a
```

Dieser IMAGE Formatstring ist aus zwei Feldspezifikatoren zusammengesetzt: Der erste spezifiziert Ziffern, Komma und Dezimalpunkt; der zweite spezifiziert 10 Zeichen Text.

Jedem Datenelement in der DISP USING oder PRINT USING Liste muß ein entsprechender Zahlen- oder Stringfeldspezifikator zugeordnet sein.

Wenn Sie auf einen externen Drucker ausgeben wollen, muß dieser mit der Anweisung PRINTER IS adressiert werden. Wenn keine PRINTER IS Einheit spezifiziert ist, werden alle Druckausgaben auf die Anzeige des HP-75 und andere DISPLAY IS Einheiten geleitet.

Begrenzer (, , /)

Begrenzer dienen zur Trennung von Feldspezifikatoren:

- Ein Komma (,) trennt zwei Spezifikatoren in der Anweisung IMAGE.
- Ein Schrägstrich (/) kann zur Trennung von zwei Stringspezifikatoren benutzt werden; seine Hauptaufgabe besteht aber darin, einen Wagenrücklauf/Zeilenvorschub (CR/LF) zu erzeugen. Der Schrägstrich kann mit einem Komma von anderen Spezifikatoren getrennt werden.

Beispiel:

```
 10 DISP USING 20
●20 IMAGE 'Kosten' ,3/,'Rabatt'
```

3/ ist gleichwertig mit ///. Die Kommata sind nicht erforderlich.

Ausgeführt als:

```
Kosten



Rabatt
```

Zeigt `Kosten` an und führt den ersten CR/LF durch.

Leere Anzeige; zweiter CR/LF.

Leere Anzeige; dritter CR/LF.

Zeigt `Rabatt` an.

Die Symbole `3/` deuten an, daß zwischen den Anzeigen der beiden Strings drei Wagenrückläufe/Zeilenvorschübe durchgeführt werden sollen. Daher werden zwei Leerzeilen angezeigt. (Auf einem Drucker würden zwischen `Kosten` und `Rabatt` zwei Leerzeilen ausgedruckt werden.)

Die folgende `IMAGE` Anweisung gibt vor `Kosten` drei Leerzeilen aus:

```
20 IMAGE 3/ 'Kosten'
```

Falls *n*/ im Formatstring am Anfang steht, werden *n* Leerzeilen ausgegeben. Falls *n*/ einem anderen Feldspezifikator folgt, werden *n* – 1 Leerzeilen ausgegeben.

Beachten Sie, daß die Anweisungen `DISP USING` und `PRINT USING` am Ende der jeweiligen Ausgabeliste einen Wagenrücklauf/Zeilenvorschub erzeugen. Wie in den Anweisungen `DISP` und `PRINT` kann der CR/LF auch in `DISP USING` und `PRINT USING` mit einem Semikolon unterdrückt werden.

Leerzeichen (x, X)

`x` (oder `X`) Spezifiziert ein Leerzeichen.

Eine vor `x` stehende Zahl spezifiziert die Anzahl der Leerzeichen; `4x` zum Beispiel bedeutet vier Leerzeichen (wie auch `xxxx`).

Spezifizieren von Strings (' ', " ", a, A)

Text kann auf zwei Weisen spezifiziert werden:

`' '` (oder `" "`) Text in Anführungszeichen wird gedruckt oder angezeigt, wie er eingegeben wurde, ohne das äußere Paar Anführungszeichen. Sie können Literale sowohl in der jeweiligen Ausgabeliste als auch in der `IMAGE` Anweisung spezifizieren.

Beispiel:

```
30 IMAGE '^^',8x,'Ergebnisse',8x
,'^^'
40 DISP USING 30
```

Mit `8x` werden acht Leerzeichen ausgegeben.

Die Anordnung der `IMAGE` Anweisung in Bezug auf `DISP USING` spielt keine Rolle.

Wird ausgeführt als:

```
^^        Ergebnisse        ^^
```

a (oder A) Spezifiziert ein einzelnes Zeichen. Eine a vorangestellte Zahl spezifiziert die Anzahl von Zeichen, die gedruckt oder angezeigt werden soll. Mit dem Stringspezifikator a wird jeder Text linksbündig angeordnet.

Die obenstehenden Beispiele könnten auch so aussehen:

```
 50 A$='Ergebnisse'
•60 IMAGE '^^',8x,10a,8x,'^^'
 70 DISP USING 60 ; A$
```

10a spezifiziert ein aus zehn Zeichen Text zusammengesetztes Feld.

Oder so:

```
 80 A$='Ergebnisse'
•90 IMAGE aa,8x,10a,8x,aa
 100 DISP USING 90 ; '^^',A$,'^^'
```

aa kann auch als 2a dargestellt werden.

Wenn das Stringelement in der Ausgabeliste länger als die spezifizierte Anzahl von Zeichen ist, wird der String verkürzt.

Beispiel:

```
 110 DISP USING 120 ; 'wohnort'
•120 IMAGE 6a
```

Es werden nur maximal 6 Zeichen ausgegeben.

Ausgeführt als:

```
wohnor
```

Wenn das Ausgabeelement kürzer ist, wird der Rest des Feldes mit Leerzeichen aufgefüllt. Die folgende IMAGE Anweisung gibt ein Feld mit 80 Zeichen aus.

```
120 IMAGE 80a
```

Der String wohnort wird von 73 (80–7) Leerstellen gefolgt.

Spezifizieren von Zahlen

Es stehen Ihnen eine Vielzahl von Symbolen zur Spezifikation von Zahlen zur Verfügung: Ziffernsymbole, Vorzeichensymbole, Dezimaltrennsymbole, Zifferntrennsymbole und ein Exponentialsymbol.

Ziffernsymbole (d, D, z, Z, *)

d (oder D) Spezifiziert eine Ziffernposition. Eine Zahl vor dem d gibt die Anzahl von Ziffernstellen vor. Wenn die Anzahl von d's links des Dezimaltrennzeichens ein größeres Feld als das numerische Datenelement spezifizieren, dann erscheint das Datenelement im Feld rechtsbündig, und führende Nullen werden durch Leerstellen ersetzt. Wenn die Anzahl von d's rechts des Dezimaltrennzeichens ein größeres Feld als das numerische Datenelement spezifizieren, dann erscheint das Datenelement im Feld linksbündig mit nachlaufenden Nullen. Falls der gebrochene Anteil des Datenelements größer als die Anzahl von d's rechts des Dezimaltrennzeichens ist, wird der Wert gerundet, um in das entsprechende Feld zu passen.

Beispiel:

```
130 DISP USING 140 ; 250,25.5
140 IMAGE 5d,2x,dd.dd
```

Ausgeführt als:

```
  250  25.50
```

Beachten Sie, daß D und d die einzigen Ziffernsymbole sind, die Ziffern *rechts* eines Dezimaltrennzeichens spezifizieren können. Um zum Beispiel einen Dezimalpunkt und zwei Ziffern zu spezifizieren, müssen Sie die Symbole .dd, .2d, .DD usw. benutzen.

z (oder Z) Spezifiziert eine Ziffernposition – führende Nullen werden durch Nullen als Füllzeichen ersetzt. Ein z kann nicht rechts eines Dezimaltrennzeichens benutzt werden. Eine Zahl vor z spezifiziert wieder die Anzahl von Ziffernstellen.

Beispiel:

```
150 DISP USING 160 ; 256,321
160 IMAGE 5z,2x,zzzzz
```

Wird ausgeführt als:

```
00256  00321
```

* Ein Stern spezifiziert ebenfalls eine Ziffernposition, aber führende Nullen werden durch Sterne als Füllzeichen ersetzt. Sie können rechts von einem Dezimaltrennsymbol keinen * verwenden. Eine Zahl vor * bestimmt die Anzahl der Sterne.

Beispiel:

```
170 IMAGE 5*,2X,5Z,2X,5D
180 DISP USING 170 ; 99,77,55
```

Wird ausgeführt als:

```
***99  00077     55
```

Wie Sie sehen, kann der ganzzahlige Anteil einer Zahl mit jedem der Ziffernsymbole *, z oder d spezifiziert werden. Sie können diese Symbole jedoch in einem IMAGE Format nicht willkürlich mischen. Wenn zum Beispiel ein d zur Spezifikation einer Ziffernposition einer Zahl benutzt wird, muß die gesamte Zahl mit d's spezifiziert werden; nur das Ziffernsymbol für die erste Vorkommastelle kann unabhängig davon auch ein z sein.

Beispiel:

```
190 DISP USING 200 ; 357,972
200 IMAGE DDZZ,2X,D*ZZ*
```

Eine unzulässige Anweisung.

Die obige IMAGE Anweisung enthält zwei ungültige Spezifikatoren (DDZZ und D*ZZZ*) und erzeugt Fehler 52 – invalid IMAGE, sobald die DISP USING Anweisung ausgeführt wird. Die folgenden IMAGE Strings sind jedoch zulässig:

```
200 IMAGE dddz,2x,*****z
```

Ausgeführt als:

```
357   ***972
```

Immer wenn ein IMAGE Feldspezifikator ein unverständliches Symbol enthält, erhalten Sie bei der Ausführung der dazugehörenden DISP USING oder PRINT USING Anweisung Fehler 52 – invalid IMAGE, und die Ausführung bleibt in der Zeile, die die DISP USING bzw. PRINT USING Anweisung enthält, stehen. Siehe auch Seite 247, «Überlauf numerischer Felder».

Dezimaltrennsymbole (., r, R)

Ein Dezimaltrennzeichen trennt den ganzzahligen Anteil einer Zahl von ihrem gebrochenen Anteil. In Europa wird gewöhnlich das Komma (34,7) als Dezimaltrennzeichen verwendet, in den U.S.A. ist der Punkt üblich. *In einem numerischen Spezifikator darf nicht mehr als ein Dezimaltrennzeichen auftreten.* Die Ziffern rechts des Dezimaltrennzeichens können nur mit dem Symbol d spezifiziert werden.

.	Spezifiziert einen Dezimalpunkt an diese Stelle.
r (oder R)	Spezifiziert das Komma als Dezimaltrennsymbol an dieser Stelle.

Beispiele:

```
210 DISP USING 220 ; 473.1,25.39
2,76.5
220 IMAGE ddd.dd,2x,***.ddd,2x,z
zzrdd
230 IMAGE ddd.ddd,4x,3z.3d,4x,.d
d
240 DISP USING 230 ; .756,99.99,
.879
```

Wird ausgeführt als:

```
473.10   *25.392   076,50
   .756     0.99     .88
```

Beachten Sie, daß .879 auf .88 gerundet wurde, da nur zwei Ziffern rechts des Dezimalpunkts spezifiziert worden sind.

Vorzeichensymbole (s, S, m, M)

Zwei Vorzeichensymbole steuern die Ausgabe der Vorzeichen + und −. In einem Zahlenspezifikator kann maximal ein Vorzeichensymbol auftreten. Wenn kein Vorzeichensymbol spezifiziert worden ist, belegt ein Minuszeichen gegebenenfalls eine Ziffernposition.

s (oder S)	Spezifiziert die Ausgabe eines Vorzeichens: + bei positiver Zahl, − bei negativer Zahl.
m (oder M)	Spezifiziert die Ausgabe eines Vorzeichens: − bei negativer Zahl, Leerzeichen bei positiver Zahl.

Beispiel:

```
250 DISP USING 260 ; -47.2,-.51,
33.5,38.12
260 IMAGE mdd.dd,2x,szz.dd,2x,sz
z.dd,2x,mzz.dd
```

Ausgeführt als:

```
-47.20   -00.51   +33.50    38.12
```

Das Vorzeichen «gleitet» mit der Zahl; zum Beispiel:

```
270 DISP USING 280 ; -5,6,-.07
280 IMAGE sddd.d,s3d.d,m2d.dd
```

Ausgeführt als:

```
-5.0   +6.0   -.07
```

In den obenstehenden Beispielen erscheinen die Vorzeichen unmittelbar links von der Zahl. Wenn Sie in Ihrem Format eines der Symbole `Z` oder `*` benutzen, erscheint das Vorzeichen links von führenden Nullen oder Sternen.

Zifferntrennsymbole (`C`, `C`, `P`, `P`)

Mit Dezimaltrennzeichen werden große Zahlen zur besseren Lesbarkeit in Zifferngruppen unterteilt (im allgemeinen drei Ziffern pro Gruppe). In Europa wird gewöhnlich der Punkt verwendet; in den U.S.A. steht dafür das Komma.

- `C` (oder `C`) Spezifiziert ein Komma als Trennzeichen an der gegebenen Stelle.
- `P` (oder `P`) Spezifiziert einen Punkt als Trennzeichen an der gegebenen Stelle.

Das Dezimaltrennsymbol wird nur ausgegeben, wenn zuvor schon eine Ziffer ausgegeben wurde; es sollte zwischen zwei Ziffern stehen. Wenn mit `Z` führende Nullen erzeugt werden, werden diese als Ziffern betrachtet und enthalten Zifferntrennsymbole. Ein Formatstring, der aus führenden Sternen besteht, kann Zifferntrennsymbole enthalten. Wenn aber nicht zu beiden Seiten des Zifferntrennsymbols Ziffern ausgegeben werden, wird der Separator durch einen Stern ersetzt.

Beispiele:

```
290 DISP USING 300 ; 25613.92,27
.96,71.5
300 IMAGE 3dc3d.dd,2x,zc3z.dd,2x,
3dc3d.dd
310 DISP USING 320 ; 9999.99
320 IMAGE dpdddrdddd
```

Ausgeführt als:

```
 25,613.92  0,027.96         71.50
9,999,9900
```

Das Exponentiationssymbol (e, E)

e (oder E) Numerische Felder, die ein e oder E enthalten, werden in Exponentialdarstellung ausgegeben. Der HP-75 drückt den Exponenten dreistellig mit Vorzeichen (+ oder −) aus. Dem Symbol e muß zumindest eine Ziffer vorausgehen.

Beispiele:

```
330 DISP USING 340 ; 157.24
340 IMAGE d.ddde
350 DISP USING 360 ; 5.762
360 IMAGE ddd.dde
```

Wird ausgeführt als:

```
1.572E+002
576.20E-002
```

Der HP-75 ergänzt soviele nachlaufende Nullen, bis die für die Mantisse spezifizierte Ziffernzahl erreicht ist und paßt den Exponenten an die Mantisse an.

Beispiel:

```
350 DISP USING 360 ; 3
360 IMAGE 25de
```

Wird ausgeführt als:

```
300000000000000000000000000E-024
```

Beachten Sie, daß nach dem Exponentialzeichen e kein Dezimaltrennzeichen oder Vorzeichensymbol mehr spezifiziert werden darf; Exponenten werden immer dreistellig (mit führenden Nullen als Platzhaltern) mit vorgestelltem Vorzeichen und ohne Dezimalpunkt dargestellt.

Kompaktfeld-Spezifikatoren (k, K)

Das Symbol k wird zur Spezifikation eines ganzen Feldes für eine Zahl oder einen Zeichenstring benutzt. Die Datenelemente werden in kompakter Form, ohne führende oder nachlaufende Nullen, ausgegeben.

Beispiel:

```
 370 DISP USING 380 ; 'ABC',415,'
 DEF',.01
•380 IMAGE k,k,k,k
```

Spezifiziert vier Kompaktfeldausgaben (kann *nicht* als 4k abgekürzt werden).

Wird ausgeführt als:

```
ABC415DEF.01
```

Replikation ( )

Viele der als IMAGE Spezifikatoren benutzten Symbole können mehrere aufeinanderfolgende Stellen spezifizieren, wenn ihnen eine Zahl von 1 bis 9999 unmittelbar vorangestellt wird. Sie haben dazu schon einige Beispiele gesehen; die folgenden Anweisungen spezifizieren alle das gleiche Ausgabeformat:

```
390 IMAGE ddd.dd
400 IMAGE d2d.2d
410 IMAGE 3d.dd
420 IMAGE 3d.2d
```

Die folgenden Symbole können wiederholt werden: x, d, z, *, a und /.

Weiterhin kann auch ein vollständiger Spezifikator oder eine Gruppe von Spezifikatoren wiederholt werden, indem diese einfach in Klammern eingeschlossen werden, denen eine ganze Zahl von 1 bis 9999 unmittelbar vorausgeht.

Beispiele:

```
430 IMAGE dd,d,6(ddd,dd)
440 IMAGE 4z,d,3(6x,7*,d,2(2x,d)
)
```

Die Spezifikation von 3(dd) ist gleichwertig mit dd,dd,dd, die Anweisung 440 ist also äquivalent zu:

```
440 IMAGE 4z,d,6x,7*,d,2x,d,2x,d
,6x,7*,d,2x,d,2x,d,6x,7*,d,2x,d,
2x,d
```

Innere Klammern werden bei jeder Wiederholung der äußeren Klammer repliziert.

Auf diese Weise kann auch k repliziert werden:

```
450 IMAGE 4(k)
```

Gleichwertig zu k,k,k,k.

Sie können mit verschachtelter Klammerung weitere Wiederholungsebenen spezifizieren.

Wiederbenutzung des IMAGE Formatstrings

Wenn ein Formatstring zu Ende ist, bevor alle Datenelemente einer Ausgabeliste ausgegeben worden sind, wird er wieder von vorn abgearbeitet.

Beispiel:

```
 460 DISP USING 470 ; 25,71,99.9,
 14,23
•470 IMAGE ddd.dd
```

Diese IMAGE Anweisung wird dreimal benutzt, um die drei Datenelemente in der DISP USING Liste auszugeben.

Ausgeführt als:

```
25.71 99.90 14.23
```

Überlauf numerischer Felder

Wenn eine Zahl *links* des Dezimalpunktes mehr Ziffern aufweist, als der Feldspezifikator zur Verfügung stellt, entsteht eine Überlauf-Bedingung.

Beispiel:

```
480 DISP USING 490 ; 336.71,-14.
3
490 IMAGE 2x, dd.dd
```

Die Zahlen, 336.71 und – 14.3, erzeugen bei der Verwendung des Spezifikators `dd.dd` einen Überlauf, da beide Zahlen drei Positionen links des Dezimalpunktes benötigen. (Erinnern Sie sich, daß ein nicht mit `S` explizit spezifiziertes Minuszeichen eine Ziffernstelle belegt.) Bei `DEFAULT ON` erhalten Sie Warnung 2 – `num too large`, es werden nur die mit `P` oder `C` spezifizierten Dezimaltrennsymbole und die mit `M` oder `S` spezifizierten oder mit `*` oder `Z` implizierten Vorzeichen ausgegeben, und die Programmausführung wird fortgesetzt. Bei `DEFAULT OFF` meldet der HP-75 einen Fehler und stoppt die Ausführung. Geben Sie mehr Ziffernpositionen ein, um das Ausgabeformat zu korrigieren:

```
490 IMAGE 2x,ddd.dd
```

Gibt drei Ziffernpositionen links des Dezimalpunktes vor.

Wird ausgeführt als:

```
   336.71 -14.30
```

Wenn ein Zeichenstring mehr Zeichen besitzt als mit den Zeichensymbolen `A` spezifiziert wurde, wird der String nach der angegebenen Anzahl von Zeichen abgeschnitten, ohne daß ein Fehler auftritt.

Formatierung in den Anweisungen `DISP USING` und `PRINT USING`

Innerhalb der Anweisungen `DISP USING` und `PRINT USING` können Sie das Ausgabeformat auch direkt spezifizieren:

```
DISP USING ' Formatstring ' [; Ausgabeliste ]
```

```
PRINT USING ' Formatstring ' [; Ausgabeliste ]
```

Anstatt die Zeilennummer einer `IMAGE` Anweisung anzugeben, können Sie in den Anweisungen `DISP USING` und `PRINT USING` vor der Ausgabeliste einen Formatstring spezifizieren. Der Formatstring kann ein in Anführungszeichen eingeschlossener String, eine Stringvariable oder ein beliebiger Stringausdruck, der das Format spezifiziert, sein.

Beispiele:

```
 500 DISP USING '4d.dd,2x,**z.ddd
  ' ; 1473.25.39
 510 DISP USING '3d.2d' ; 310.12,
 56,42.5
•520 F$='3dc3d.2d'
 530 DISP USING F$ ; 25613.92,279
 6
```

Denken Sie daran, den String zu dimensionieren, wenn er länger als 32 Zeichen ist.

Ausgeführt als:

```
1473.00  *25.390
310.12 56.00 42.50
 25,613.92  2,796.00
```

Der in F$ gespeicherte Spezifikator wird zweimal benutzt.

Beachten Sie, daß in einer DISP USING oder PRINT USING Anweisung der *gesamte* Formatstring mit einem äußeren Paar von Anführungszeichen begrenzt werden muß. Die folgende Anweisung DISP USING zum Beispiel enthält einen unvollständigen String und ist nicht erlaubt.

```
 >540 disp using 'Name',2x,10a,'A
lter',3d ; 'Charley',43
```

RTN ergibt Fehler 84 – extra characters.

Der Fehler entsteht nach dem zweiten Anführungszeichen, da mehrfache Formatstrings in einzelnen Anweisungen DISP USING nicht erlaubt sind. Das gleiche Format könnte auf verschiedene Weisen spezifiziert werden:

```
540 DISP USING "'Name',2x,10a,"A
lter',3d" ; 'Charley',43
```

Der gesamte Formatstring steht in Anführungszeichen.

```
550 DISP USING '4a,2x,10a,5a,3d'
 ; 'Name','Charley','Alter',43
```

Die drei Strings werden durch die Zeichenspezifikatoren a dargestellt.

```
560 DISO USING 570 ; 'Charley',4
 3
570 IMAGE 'Name',2x,10a,'Alter',
3d
```

Anwendung einer DISP USING/IMAGE Kombination.

Die Ausführung jeweils:

```
Name  Charley   Alter 43
```

Wenn der Formatstring eines DISP USING oder PRINT USING Ausgabeelements ein unzulässiges oder ein unerkanntes Zeichen enthält, entsteht bei der Ausführung der Anweisung Fehler 52 – invalid IMAGE.

DISP USING und PRINT USING Anweisungen, die Formatstrings enthalten, können über das Tastenfeld ausgeführt werden; sobald diese Anweisungen aber auf eine IMAGE Anweisung Bezug nehmen, können sie nur in Programmen ausgeführt werden.

Zusammenfassung der Feldspezifikatoren

Wir fassen die Feldspezifikatoren und deren Verwendung noch einmal zusammen. IMAGE Spezifikatoren können mit Kommata oder Querstrichen begrenzt werden.

IMAGE Symbol	Ausgabe	Beschreibung	Wiederholung möglich
x,X	Leerstelle	Spezifiziert eine Leerstelle zwischen Datenelementen.	ja
' ',""	Literal	Zur Begrenzung von Strings in Formatstrings oder ganzen Spezifikatoren in DISP USING und PRINT USING Anweisungen.	nein
a,A	Zeichen	Spezifiziert die Position eines Zeichens, Text steht linksbündig.	ja
d,D	Ziffer	Spezifiziert die Position einer Ziffer links oder rechts des Dezimaltrennsymbols; führende Leerstellen, nachlaufende Nullen.	ja
z,Z	Ziffer	Spezifiziert die Position einer Ziffer links des Dezimaltrennsymbols; führende Nullen.	ja
*	Ziffer	Spezifiziert die Position einer Ziffer links des Dezimaltrennsymbols; führende Schutzsterne.	ja
s,S	Vorzeichen	Spezifiziert ein Vorzeichen, + oder −.	nein
m,M	Vorzeichen	Spezifiziert ein Vorzeichen, Leerstelle oder −.	nein
e,E	Exponentialdarstellung	Gibt Zahlen mit Exponenten E, Vorzeichen und drei Stellen aus.	nein
.	Dezimalpunkt	Gibt einen Dezimalpunkt als Dezimaltrennsymbol an dieser Stelle aus.	nein
r,R	Komma	Gibt ein Komma als Dezimaltrennsymbol an dieser Stelle aus.	nein
c,C	Komma	Gibt ein Komma als Zifferntrennsymbol an dieser Stelle aus.	nein
p,P	Punkt	Gibt einen Punkt als Zifferntrenner an dieser Stelle aus.	nein
()	Feld	Ermöglicht die Wiederholung der eingeschlossenen Feldspezifikatoren.	ja
k,K	Kompakt-Daten	Strings und Zahlen werden ohne führende oder nachlaufende Leerstellen ausgegeben.	nein
/	CR/LF	Erzeugt einen Wagenrücklauf/Zeilenvorschub; kann auch Datenelemente begrenzen.	ja

Die IMAGE Anweisung ist ein sehr vielseitiges und komplexes Werkzeug zur Kontrolle von Anzeigeformaten. Sie sollten IMAGE Anweisungen sehr sorgfältig spezifizieren. Unpassende Spezifikationen werden nicht immer als Fehler gemeldet; die Ausgabe bei falschen oder unpassenden Formaten kann jedoch unerwartete Ergebnisse zeigen. Wie Sie in diesem Abschnitt gesehen haben, können Sie mit der IMAGE Anweisung das Format Ihrer Ausgaben vollständig kontrollieren. Vergegenwärtigen Sie sich beim Entwurf Ihrer IMAGE Anweisungen den Wertebereich Ihrer Ausgabedaten, und spezifizieren Sie Formate, die den gesamten Wertebereich umfassen. Die Hauptfaktoren, die bei der Formatierung der Ausgabe berücksichtigt werden müssen, sind die Länge der Anzeige- oder Druckzeile, die durch die DISPLAY IS und PRINTER IS Einheiten vorbestimmt ist, die derzeitigen Werte von WIDTH und PWIDTH, und die Anzahl signifikanter Stellen in den Ausgabedaten. Allgemein sollte so formatiert werden, daß die in eine Zeile auszugebende Information die maximal zulässige Anzahl von Zeichen pro Zeile nicht überschreitet.

Die Anweisungen DISP USING und PRINT USING ignorieren die Formatinformation am Ende des Formatstrings, wenn die USING Liste mit einem Semikolon endet.

Beispiel:

```
100 IMAGE dd.dd, ' temp.'
200 DISP USING 100 ; 32;
```

Die IMAGE-Anweisung enthält einen numerischen Spezifikator und einen String in Anführungszeichen.

Wird ausgeführt als:

```
32.00
```

Das Literal ' temp ' wird nicht angezeigt.

Abschnitt 17

Fehlersuche

Inhalt

Einführung

Bei der Programmierung sind Fehler in der Regel unvermeidlich. Man unterscheidet vier Fehlerarten: Syntaxfehler (sprachbezogen), Initialisierungsfehler (wie zum Beispiel fehlende Zeilennummern bei GOTO oder doppelte Variablendeklaration), Laufzeitfehler und Logikfehler. Der HP-75 prüft bei der Eingabe auf richtige Syntax; Syntaxfehler werden also sofort offenbar. Initialisierungsfehler werden bei der Initialisierung des Programms angezeigt, und können ebenfalls schnell behoben werden. Laufzeitfehler und Logikfehler sind schwieriger zu bereinigen, da sie oft durch Fehler in der Programmstruktur, wie einer Wertzuweisung an eine Variable an der falschen Stelle, Nichtaussortieren unpassender Eingabewerte oder Verwendung des falschen Algorithmus entstehen.

Der HP-75 besitzt eine Reihe von Einrichtungen zur Suche und Kontrolle von Laufzeitfehlern. Mit den Befehlen TRACE FLOW und TRACE VARS können Sie Programmverzweigungen und Wertänderungen bei Variablen nachvollziehen. Die Tastenkombination [SHIFT] [RUN] ermöglicht Ihnen die Abarbeitung des Programms in Einzelschritten. Mit Hilfe der Anweisung ON ERROR können Sie Ihre eigenen Fehlerbehandlungsroutinen entwerfen.

Programmverfolgung

Bei der Suche nach Strukturfehlern ist es hilfreich, die Reihenfolge der Ausführung von Anweisungen (mit TRACE FLOW) und der Wertzuweisung an Variablen (mit TRACE VARS) zu verfolgen. Alle Trace-Ausgaben werden entsprechend der momentanen Verzögerungsrate auf die Anzeige des HP-75 und die momentanen DISPLAY IS Einheiten geleitet.

Verfolgen von Verzweigungen (TRACE FLOW)

Die Anweisung TRACE FLOW wird benutzt, um die Reihenfolge der Ausführung der Anweisungen im ganzen Programm oder in Teilen davon zu verfolgen.

```
TRACE FLOW
```

Wenn die Reihenfolge der Programmausführung der Reihenfolge der Zeilennummern folgt, wird nichts angezeigt. Aber immer wenn eine Verzweigung im Programm erfolgt, werden die Zeilennummer, an der die Verzweigung erfolgt, und die Zeilennummer, zu der das Programm verzweigt, in der folgenden Form angezeigt:

```
TRACE Zeilennummer TO Zeilennummer
```

TRACE FLOW verfolgt jedoch keine CALL Anweisungen. Wenn ein Programm ein anderes aufruft, wird keine Tracemeldung angezeigt, weder beim Aufruf noch bei der Rückkehr zum aufrufenden Programm.

Verfolgen von Variablen (TRACE VARS)

Der Befehl TRACE VARS ermöglicht Ihnen, die Wertänderungen von Variablen zu verfolgen.

```
TRACE VARS
```

Immer wenn einer Programmvariablen ein Wert zugewiesen wird, gibt die Trace-Ausgabe an, wo die Wertzuweisung stattgefunden hat und:

- den Namen und den zugewiesenen Wert bei einer einfachen numerischen Variablen. Beispiel:

```
Trace line 10 B=2
```

- den Namen einer Stringvariablen. Beispiel:

```
Trace line 20 S$
```

- den Namen, den Index/die Indices und den zugewiesenen Wert eines bestimmten Feldelementes. Beispiel:

```
Trace line 30 C(2,3)=45
```

- den Namen eines Feldes in der Anweisung READ # in der Form A() oder C(,), und den Wert des Feldes. Beispiel:

```
Trace line 400 A()=75
```

Mit TRACE VARS können Sie prüfen, ob Variablen an der richtigen Stelle die richtigen Werte zugewiesen bekommen.

Aufheben von Trace-Operationen (TRACE OFF)

Trace-Operationen werden mit der Anweisung TRACE OFF aufgehoben:

```
TRACE OFF
```

Mit TRACE OFF können Sie einen TRACE FLOW Zustand, einen TRACE VARS Zustand, oder beides aufheben.

Verwenden von TRACE Befehlen

TRACE Befehle können programmiert oder direkt über das Tastenfeld ausgeführt werden. Da die Befehle global wirken, bleibt der momentane Trace-Zustand wirksam, bis ein anderer Trace-Befehl ausgeführt wird. Die Befehle TRACE VARS und TRACE FLOW sind unabhängig voneinander, sie können jederzeit einzeln oder gleichzeitig wirksam sein.

Setzen Sie den TRACE FLOW oder den TRACE VARS Zustand oder beide, wenn Sie die Programmausführung verfolgen wollen, und starten Sie das Programm.

Beispiel: Verfolgen Sie die Verzweigungen des folgenden Programms zur Umwandlung einer Zahlenbasis bei dessen Berechnung der Oktaldarstellung (Basis 8) der Binärzahl 10101010.

```
 10 ! --Zahlenbasis-Umwandlung--
 20 DIM B$[16],F$[24]
•30 B$='0123456789ABCDEF'
 40 INPUT 'Eingabebasis, Ausgabe
 basis: ';B1,B2
 50 DISP 'Zahl in Basis '&STR$(B
 1);
•60 INPUT J$
 70 N=0
 80 FOR J=1 TO LEN(J$)
 90 P=POS(B$[1,B1],UPRC$(J$[J,J])
 )
•100 IF P=0 THEN 260
•110 N=B1*N+P-1
 120 NEXT J
•130 F$=''
•140 N=N/B2
•150 P=B2*FP(N)+1
•160 F$=F$&B$[P,P]
 170 N=INT(N)
•180 IF N#0 THEN 140
•190 DISP J$;' Basis';B1
 200 WAIT 2
 210 FOR K=LEN(F$) TO 1 STEP -1
 220 DISP F$[K,K];
 230 NEXT K
 240 DISP ' Basis';B2
 250 STOP
 260 BEEP
 270 DISP J$[J,J];'unzulaessig i
 n Basis';B1
 280 WAIT 2
•290 GOTO 50
```

Zuweisung des Prüfstrings.

Zahleneingabe in der ersten Basis.

Springt bei nicht erlaubtem Zeichen nach Zeile 260.
Berechnet das Äquivalent zur Basis 10.

Initialisiert den Ausgabestring.
Verschiebt um eine Stelle nach rechts.
Ruft ein Zeichen ab.
Kehrt die Reihenfolge der Zeichen im String um.

Prüft, ob die Berechnung beendet ist.
Zeigt die ursprüngliche Eingabe an.

Zeigt das Ergebnis an.

Nächste Eingabe.

Setzen Sie nach der Eingabe des Programms eine TRACE FLOW Bedingung vom Tastenfeld aus: tasten Sie dazu `trace flow` RTN ein. Drücken Sie danach RUN:

```
Eingabebasis, Ausgabebasis: ■
Zahl in Basis 2?


Trace line 120 to 80
Trace line 120 to 80
Trace line 120 to 80
Trace line 120 to 80
Trace line 120 to 80
Trace line 120 to 80
Trace line 120 to 80

Trace line 180 to 140
Trace line 180 to 140

10101010 Basis 2

2 Trace line 230 to 210
5 Trace line 230 to 210
2 Basis 8
```

Geben Sie 2,8 RTN ein, um binär in oktal umzuwandeln. Geben Sie für dieses Beispiel 10101010 RTN ein.

Das Programm führt die gleiche Verzweigung von 120 nach 80 siebenmal hintereinander aus, um die Originalzahl in ihr dezimales Äquivalent zu verwandeln.

Dann führt das Programm die Zeilen 140 bis 180 dreimal aus (zwei Verzweigungen), um den Zeichenstring des Oktalwertes zu erzeugen.

Zeile 190 zeigt die Originalzahl und -basis an.

Schließlich wird der String 252 Zeichen für Zeichen mit der FOR...NEXT Schleife von Zeile 210 bis 230 angezeigt.

Das Programm verläßt die FOR...NEXT Schleife in den Anweisungen 80 bis 120, sobald der Dezimalwert der eingegebenen Zahl in der Variablen N abgespeichert wurde. Beachten Sie, daß die Ausführung nicht zu den FOR Anweisungen in den Zeilen 80 und 210 zurückkehrt, um die Anweisung FOR wieder durchzuführen (dies würde den Schleifenzähler zurücksetzen), sondern um die Schleifenaustrittsbedingung abzufragen.

Sie können jeden der TRACE Befehle innerhalb eines Programms verwenden. Fügen Sie zum Beispiel die folgenden Zeilen in Ihr Zahlenbasis-Umwandlungsprogramm ein:

```
75 TRACE VARS

125 TRACE OFF
```

Führen Sie nun TRACE OFF über das Tastenfeld aus, um den vorhergehenden TRACE FLOW Zustand aufzuheben, und starten Sie das Programm, um den Dezimalwert von FF_{16} zu berechnen. (Erinnern Sie sich, daß die Ziffern zur Basis 16 Werte von 0 bis F annehmen, die den Werten 0 bis 15 entsprechen.)

```
Eingabebasis,Ausgabebasis: ■
Zahl in Basis 16?■
Trace line 80 J=1
Trace line 90 P=16
Trace line 110 N=15
Trace line 120 J=2
Trace line 90 P=16
Trace line 110 N=255
Trace line 120 J=3

ff Basis 16
255 Basis 10
```

Geben Sie 16,10 ein und drücken Sie RTN. Geben Sie f f RTN ein.

Da die Werte der Variablen F$ und B$ zwischen den Anweisungen 75 und 125 nicht geändert werden, erscheinen sie nicht in der TRACE VARS Liste.

Denken Sie daran, daß eine TRACE-Bedingung solange wirksam bleibt, bis sie mit einem anderen TRACE Befehl, entweder im Programm oder über das Tastenfeld, geändert wird.

Einzelschritt-Ausführung (SHIFT RUN)

Die Tastenkombination SHIFT RUN dient zur Abarbeitung des Programms in Einzelschritten. Wenn Sie SHIFT RUN drücken, wird die als nächste auszuführende Zeile angezeigt; beim Loslassen der Tasten werden alle Anweisungen der Zeile ausgeführt.

Beispiel: Führen Sie das Programm zur Zahlenbasisumwandlung von der ersten INPUT Anweisung an in Einzelschritten aus, um die Umwandlung von 3_{10} ins Binärsystem mitzuvollziehen. Heben Sie zuerst den Befehl TRACE VARS in Zeile 75 auf, damit die Anweisungen ohne Trace-Ausgaben betrachtet werden können. Initialisieren Sie dann das Programm mit RUN:

```
Eingabebasis,Ausgabebasis: ▮
                      PRGM
```

Unterbrechen Sie jetzt das Programm mit ATTN.

Beginnen Sie das Einzelschrittverfahren: Drücken Sie SHIFT RUN und *halten Sie die Taste* RUN *gedrückt*:

```
40 INPUT 'Eingabebasis, Ausgabeb
                      PRGM
```

Die nächste auszuführende Zeile wird solange angezeigt, wie die Taste RUN gedrückt bleibt.

Lassen Sie dann die Taste los:

```
Eingabebasis, Ausgabebasis: ▮
                      PRGM
```

Die INPUT Anweisung wird ausgeführt. Der HP-75 wartet auf Ihre Eingabe. Geben Sie 10,2 RTN ein, um dezimal in binär umzuwandeln.

Nach Ihrer Eingabe bleibt das Programm bei der auf die INPUT Anweisung folgenden Anweisung stehen. Mit SHIFT RUN können Sie diese anzeigen und ausführen:

```
50 DISP 'Zahl in Basis '&STR$(B1
                      PRGM
```

Die Programmzeile wird angezeigt, bis die Taste RUN losgelassen wird.

```
Zahl in Basis 10
```

Die DISP Anweisung wird ausgeführt.

Drücken Sie nochmals SHIFT RUN:

```
60 INPUT J$
                      PRGM
```

Als nächstes auszuführen.

Lassen Sie die Taste los:

```
? ▮
                      PRGM
```

Geben Sie 3 als Antwort auf die INPUT Anweisung ein.

Drücken Sie nochmals SHIFT RUN:

```
70 N=0
                      PRGM
```

Wenn Sie die Taste diesmal loslaßen, wird die Statusanzeige **PRGM** zurückgesetzt und die Zeile 70 bleibt in der Anzeige.

Der HP-75 erhält, nachdem Sie SHIFT RUN loslassen, die letzte Anzeige, solange die Programmausführung keine Ausgaben ins Anzeigefenster leitet.

Nach dem Loslassen der Taste können Sie mit den Tasten ← und → und den Umschalttasten SHIFT und CTL lange Programmzeilen untersuchen. Folgen Sie dem Zahlenbasisprogramm bis zum Schluß. Die Anweisungen werden in der Reihenfolge ihrer Ausführung angezeigt, einschließlich der Verzweigung von Zeile 180 zu Zeile 140. Beachten Sie, daß die Statusanzeige **PRGM**, während SHIFT RUN gedrückt wird, und während der Ausführung der Programmanweisungen gesetzt bleibt. Bei der Anweisung `WAIT` in Zeile 200 bleibt die Statusanzeige zum Beispiel für 2 Sekunden gesetzt. Sie können jederzeit `cont` RTN eingeben, um das Programm ohne Unterbrechung fortzusetzen.

Bevor Sie zum ersten Mal SHIFT RUN drücken, muß das Programm initialisiert werden; andernfalls entsteht Fehler 31 – `CONT before RUN`. Es wird empfohlen, eine `STOP` Anweisung an der Stelle einzufügen, wo Sie mit der Einzelschrittausführung beginnen wollen, zum Beispiel bei Zeile 0, wenn Sie das Programm vom Beginn an in Einzelschritten ausführen wollen, und dann RUN zu drücken. Sobald die Anweisung `STOP` die Ausführung unterbricht, können Sie von da an mit SHIFT RUN in Einzelschritten weiterarbeiten.

Es ist möglich und oft nützlich, bei der Einzelschritt-Ausführung eines Programmes die Werte von Variablen zu prüfen und zu ändern.

Angenommen, das initialisierte Programm ist gleichzeitig der zur Zeit editierte File, dann wird mit SHIFT RUN sowohl der Programmpointer *als auch* der Filepointer auf die als nächstes auszuführende Zeile (die Zeile nach der derzeitig angezeigten) gesetzt. Mit FETCH und danach RTN können Sie diese Zeile editierbereit in die Anzeige holen. Beachten Sie jedoch, daß bei der Editierung einer Zeile das Programm deallokatisiert wird.

Wenn das Programm ein anderes aufruft, können Sie mit SHIFT RUN das aufgerufene Programm in Einzelschritten, beginnend bei der Anweisung `CALL`, abarbeiten. Wenn das aufrufende Programm gleichzeitig der momentane BASIC-File ist, bleibt der Filepointer bei der `CALL` Anweisung stehen, bis die Ausführung zum aufrufenden Programm zurückkehrt.

Wenn ein Programm die Anweisung `ON TIMER #` enthält, wird der Timer bei der Einzelschrittbearbeitung dieser Zeile *nicht* gesetzt. Folglich erhalten Sie keine Timer-Unterbrechung, und die Timeranweisungen werden nicht ausgeführt.

Sie können SHIFT RUN zusammen mit Trace-Befehlen verwenden. Wenn Sie SHIFT RUN drücken, wird die Programmanweisung angezeigt; beim Loslassen wird die Anweisung ausgeführt, und eine fällige Trace-Ausgabe wird angezeigt.

Überprüfen eines angehaltenen Programms

Wenn ein Programm mit ATTN, `STOP` oder durch eine Fehlerbedingung angehalten worden ist, können Sie verschiedene Fehlersuchoperationen durchführen:

- Sie können Variablenwerte prüfen, indem Sie den Namen der Variablen gefolgt von RTN eingeben.
- Sie können den Variabeln über das Tastenfeld neue Werte zuweisen, zum Beispiel `A(5,2)=7` und `B=0`.
- Sie können Rechnerausdrücke, wie zum Beispiel `SIN(4.4)`, berechnen.
- Sie können Systembefehle ausführen, zum Beispiel `DISPLAY IS *`, `PLIST`, `DELAY`, `COPY` und `OPTION ANGLE DEGREES`.

Keine dieser Operationen hat Einfluß auf den Programmpointer, der auf die Anweisung zeigt, wo die Ausführung unterbrochen worden ist. Wenn das unterbrochene Programm der momentane BASIC-File ist, ist der Filepointer ebenfalls auf die Zeile der Unterbrechung gesetzt; andernfalls bleibt er bei der zuletzt editierten Zeile stehen.

Denken Sie daran, daß die Variablen des zur Zeit angehaltenen Programms Vorrang vor Rechnervariablen besitzen. Wenn zwei Variabeln den gleichen Namen haben, von denen einer im Programm und einer vom Tastenfeld zugewiesen worden ist, wird mit diesem Namen solange die *Programmvariable* angesprochen, bis das Programm deallokatisiert wird.

Behandlung von Laufzeitfehlern (ON ERROR, OFF ERROR)

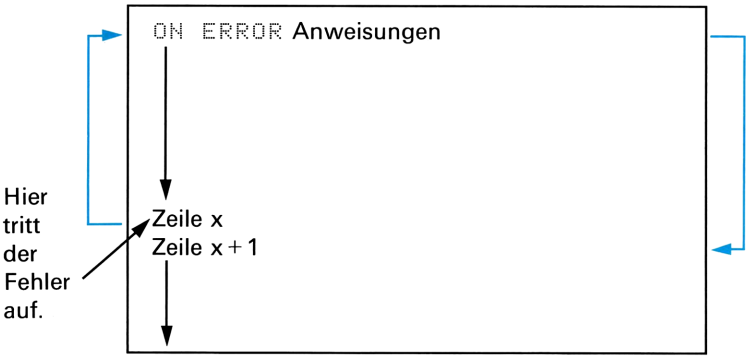
Laufzeitfehler sind die Fehler, die bei der Ausführung des Programms, nach der Initialisierung, auftreten. Dazu gehören zum Beispiel READ # Anweisungen, die den HP-75 zum Lesen über das Ende eines Datenfiles zwingen und LET Anweisungen, die einer Stringvariabeln mehr Zeichen zuweisen, als deren Länge erlaubt. Der HP-75 akzeptiert möglicherweise eine falsche Anweisung oder einen falschen Befehl beim Schreiben des Programms, entdeckt den Fehler dann jedoch bei der Ausführung. Laufzeitfehler halten die Ausführung des Programms normalerweise an. Der Befehl DEFAULT ON (Seite 89) gibt normalerweise für einige Fehler verursachende Operationen (wie die Division durch Null) zwar Ersatzwerte vor, wenn aber ON ERROR deklariert wurde, werden diese Operationen immer wie die anderen Fehler gemeldet und behandelt.

Mit der ON ERROR Anweisung können Sie die Behandlung von Laufzeitfehlern steuern, so daß Programme bestimmte Fehler «abfangen», problemorientierte Fehlermeldungen anzeigen, Ersatzwerte stellen und die Ausführung dann fortsetzen können. Programmsegmente, die Laufzeitfehler verbessern, werden Fehlerbehandlungsroutinen genannt.

ON ERROR [*zulässige Anweisung...*] *oder* [*Befehl...*]

In ON ERROR können Sie alle Anweisungen und Befehle spezifizieren, die nach THEN zulässig sind.

Wenn nach einer ON ERROR Deklaration ein Laufzeitfehler auftritt, wird die Ausführung von der den Fehler erzeugenden Anweisung an die ON ERROR Anweisung übertragen. Die Instruktion(en) der ON ERROR Anweisung werden ausgeführt, und danach wird die Programmkontrolle wieder an die erste Anweisung nach der den Fehler erzeugenden Anweisung zurückgegeben. Bei ON ERROR werden undefinierten Variablen (dies erzeugt normalerweise Warnung 7 – no value) auch keine Ersatzwerte zugewiesen.



Nach Ausführung der ON ERROR Anweisungen kehrt die Ausführung zu der auf die Fehlerbedingung folgende Zeile zurück.

Beispiel:

```
 ⋮
 1010 ON ERROR A=0 @ RESTORE
•1020 READ A
 1030 DISP @ WAIT 2
 ⋮
```

Wenn der Datenpointer hinter dem letzten Datenelement steht, bringt diese Anweisung die ON ERROR Instruktion in Zeile 1010 zur Ausführung – A wird der Wert 0 zugewiesen, der Datenpointer wird auf das erste Datenelement im File zurückgesetzt und die Programmausführung wird bei Zeile 1030 fortgesetzt.

Beachten Sie, daß die Ausführung zu der *Zeile*, nicht zu der Anweisung nach dem Fehler zurückkehrt; deshalb werden alle Anweisungen nicht ausgeführt, die in einer Zeile mit Mehrfachanweisungen einem Fehler folgen.

Es kann immer nur eine ON ERROR Deklaration gleichzeitig wirksam sein. Sie können eine ON ERROR Deklaration durch eine andere ersetzen oder mit der Anweisung OFF ERROR desaktivieren.

```
OFF ERROR
```

Wenn OFF ERROR ausgeführt wird, werden die nachfolgenden Fehler vom Betriebssystem wieder wie gewöhnlich bearbeitet.

Beispiel:

```
1010 ON ERROR A=0 @ RESTORE @ OF
F ERROR
```

Die letzte Anweisung hebt die ON ERROR Deklaration auf.

Fehlerbehandlungsroutinen (ON ERROR GOTO, ON ERROR GOSUB)

Die meisten ON ERROR Deklarationen enthalten die Anweisung GOTO oder GOSUB, um die Bearbeitung des Fehlers an eine Fehlerbehandlungsroutine irgendwo im Programm weiterzugeben. GOTO als eine ON ERROR Instruktion erzeugt eine unbedingte Verzweigung zur spezifizierten Programmzeile.

Beispiel:

```
 ⋮
 1000 S=0
 1010 ON ERROR GOTO 1050
•1020 READ A



 1030 S=S+A
 1040 GOTO 1020
•1050 OFF ERROR

 1060 DISP S
 ⋮
```

Diese Anweisung erzeugt einen Fehler, sobald der Datenpointer über das letzte Datenelement hinwegbewegt wird; danach wird die Anweisung GOTO 1050 ausgeführt.

Nach dem Fehler wird die Ausführung von hier wieder normal fortgesetzt.

Wenn durch ON ERROR GOTO nach einem Fehler eine Verzweigung erfolgt, wird die Ausführung des Programms an der spezifizierten Stelle fortgesetzt und kehrt *nicht* zu der auf die Fehlerzeile folgenden Anweisung zurück. Mit ON ERROR GOTO wird der gesamte Programmverlauf als Folge der Fehlerbedingung geändert.

Wenn die ON ERROR Anweisung eine GOSUB Anweisung enthält, wird die Ausführung an den Anfang des Unterprogramms übertragen. Sobald im Unterprogramm ein RETURN angetroffen wird, kehrt das Programm zum Ende der Anweisung ON ERROR zurück, um weitere Anweisungen auszuführen. Danach wird die Ausführung bei der auf die Fehlerzeile folgenden Zeile wieder aufgenommen. Wenn der Fehler inmitten einer Zeile mit Mehrfachanweisungen auftritt, kehrt die Ausführung ebenfalls zur nächsten Zeile – und nicht zur nächsten Anweisung – zurück.

ON ERROR wird benutzt, um Laufzeitfehler – im Gegensatz zu Syntaxfehlern oder Initialisierungsfehlern – abzufangen. Im allgemeinen wird jede Bedingung, die nach der Initialisierung eines Programms eine **ERROR** Anzeige erzeugt, eine Verzweigung zu ON ERROR Anweisungen bewirken. Beachten Sie, daß alle mathematischen Fehler, auch solche, die bei DEFAULT ON Warnungen anzeigen, zu ON ERROR Instruktionen verzweigen.

Weitere Betrachtungen

Um eine möglichst vollständige Fehlerkontrolle aufzubauen, sollten Sie versuchen, die möglichen Fehlersituationen im voraus abzuschätzen und dann entsprechende ON ERROR Deklarationen in die Bereiche zu setzen, wo der Fehler wahrscheinlich auftreten wird. Heben Sie die ON ERROR Deklaration dann nach dem Abfangen des Fehlers mit OFF ERROR auf, um nicht unerwartete Fehler mit einer unpassenden Verbesserungsroutine zu bearbeiten.

Eine ON ERROR Anweisung bleibt im Programm, das sie enthält, solange gültig, bis sie durch eine andere ON ERROR Deklaration ersetzt oder mit OFF ERROR aufgehoben wird. Die Anweisungen ON ERROR und OFF ERROR wirken lokal und beeinflussen nur das Programm, in dem sie auftreten. Wenn ein Programm ein anderes aufruft, wird die ON ERROR Anweisung kurzfristig desaktiviert, bis die Ausführung zum aufrufenden Programm zurückkehrt. Ein Versuch, ON ERROR über das Tastenfeld durchzuführen, erzeugt Fehler 88 – bad statement.

Eine ON ERROR Routine kann sich nicht selbst aufrufen. Das heißt, wenn eine ON ERROR Routine selbst einen Fehler enthält, wird die Programmausführung gestoppt.

Eine ON ERROR Unterbrechung erzeugt wie eine ON TIMER # Unterbrechung eine anstehende Rücksprungbedingung, die nach der Ausführung aller ON ERROR Instruktionen ausgeführt wird. Falls die Ursache der ON ERROR Unterbrechung Fehler 49 – GOSUB overflow – war, wurden schon zuviele anstehende Rücksprungbedingungen erzeugt. In diesem speziellen Fall meldet die ON ERROR Anweisung Fehler 29 – ON ERROR overflow – und stoppt die Ausführung, anstatt die ON ERROR Instruktion auszuführen.

Eine wichtige Form der ON ERROR Anweisung ist einfach ON ERROR RETURN. Diese Anweisung bewirkt, daß nach jedem Fehler die Ausführung mit der nächsten Programmzeile fortgesetzt wird. ON ERROR RETURN besagt im Klartext «Führe die nächste Programmzeile aus, ohne Rücksicht auf Fehler». Seien Sie jedoch im Umgang mit ON ERROR Deklarationen vorsichtig. Mit ihrer Hilfe kann ein Programm einerseits Laufzeitfehler selbst verbessern, andererseits können sie viele Fehler verstecken und das Auffinden dieser Fehler erschweren.

Fehlernummern und -zeilen (ERRN, ERRL)

Zusammen mit ON ERROR Routinen können die numerischen Funktionen ERRN und ERRL die Art und die Zeile des Auftretens eines Fehlers oder einer Warnung bestimmen. Die Information über die Zeile und die Art des Fehlers kann dann zur Steuerung der Programmausführung benutzt werden.

Die Funktion ERRN. ERRN (*Error Number = Fehlernummer*) gibt die Nummer der zuletzt aufgetretenen Fehlermeldung oder Warnmeldung zurück.

Die Funktion ERRL. ERRL (*Error Line = Fehlerzeile*) gibt die Zeilennummer des Auftretens des letzten Fehlers oder der letzten Warnung zurück.

Das folgende Programm enthält einige offensichtliche Fehler für Demonstrationszwecke.

```
 10 M=INF
 20 DISP M
•30 ON ERROR GOTO 80
 40 K=M+M
•50 OFF ERROR
 60 DISP K
 70 GOTO 120
•80 OFF ERROR


•90 IF ERRN=2 THEN 120

•100 DISP 'Fehler';ERRN;'in Zeile
 e';ERRL
 110 STOP
•120 DISP 'Zeile 120.'
```

Falls ein Fehler auftritt, wird zu Zeile 80 verzweigt.

Aufheben der ON ERROR Deklaration.

Auch hier kann eine ON ERROR Deklaration desaktiviert werden: die erste Zeile einer ON ERROR Routine.
Falls Fehler Nr.2 – num too large – auftritt, Verzweigung nach Zeile 120.
Anzeige der Fehler- und Zeilennummer, falls kein Overflow vorliegt.

Falls ein Overflow vorliegt, springt die Ausführung zu dieser Zeile.

Starten Sie nun das Programm:

```
 9.9999999999E499
Zeile 120.
```

In Zeile 40 wurde ein Overflow erzeugt (Warnung 2 – num too large); die Instruktion GOTO in der ON ERROR Anweisung wurde ausgeführt. Setzen Sie eine Verzögerungsrate von 2 Sekunden und eine TRACE Anweisung. Geben Sie delay 2 RTN und dann trace flow RTN ein, und starten Sie das Programm:

```
9.9999999999E499
Trace line 40 to 30
Trace line 30 to 80
Trace line 90 to 120
Zeile 120.
```

Der Fehler wird in Zeile 40 erzeugt.
Die ON ERROR Anweisung gibt die Programmkontrolle an Zeile 80.
ERRN ergibt 2, es erfolgt also noch eine Verzweigung.

Schreiben Sie nun die Anweisung 40 um:

```
40 K=M/0
```

Erzeugt eine Division durch Null.

Starten Sie das Programm bei gesetzter TRACE FLOW Bedingung:

```
9.9999999999E499
Trace line 40 to 30
Trace line 30 to 80
Fehler 8 in Zeile 40
```

Zeigt die Fehlernummer für Division durch 0 und die Nummer der Zeile, wo dieser Fehler auftrat, an und hält dann an.

Da das Programm initialisiert bleibt, können Sie jetzt verschiedene Überprüfungen des Programms oder Änderungen von Variablenwerten durchführen.

ERRN und ERRL sind sehr nützlich, wenn Sie die Fehlerarten, die in einem bestimmten Programmsegment zu erwarten sind, vorhersehen und einplanen können.

Anhänge

Anhang A

HP-75 Standardzubehör

Das folgende Standardzubehör liegt Ihrem HP-75 bei:

- HP-75 *Benutzerhandbuch.*
- HP-75 *Referenzhandbuch.*
- HP-75 Benutzerpaket, bestehend aus acht vorbespielten und zehn leeren Magnetkarten und einer Lesekopf-Reinigungskarte.
- Tastenfeldoverlays (1 Overlay für das numerische Tastenfeld, 1 Overlay für die HP-75 Anzeigezeichen, 3 leere Overlays).
- Zubehörbroschüre.
- Servicekarte.
- Tragetasche.
- Aufladbarer Nickel-Cadmium Batteriesatz.
- Netzadapter/Ladegerät.
- Zwei HP-IL Kabel (HP82167B) von 1 Meter Länge. (Jeder HP-IL Einheit liegt ein passendes Kabel bei.)
- Kartenhalter.

In der Zubehörbroschüre wird weiteres Zubehör, das zusätzlich für den HP-75 verfügbar ist, beschrieben. Weitere Informationen können Sie über Ihren HP-Händler erhalten.

Außerhalb der U.S.A. wenden Sie sich bitte an die nächste HP-Verkaufsniederlassung. Die Verfügbarkeit von Zubehörteilen kann unter Umständen nicht immer gewährleistet sein.

Anhang B

Information für den Benutzer

Inhalt

Seriennummer und Version des Betriebssystems (VER$)

Jeder HP-75 Computer trägt eine individuelle Seriennummer an der Unterseite des Geräts. Wir empfehlen Ihnen, diese Nummer getrennt zu vermerken. Sollte Ihr Gerät abhanden kommen, ist die Seriennummer oft bei der Wiederentdeckung und für Versicherungszwecke von Bedeutung. Hewlett-Packard führt keine Aufzeichnungen über die einzelnen Besitzer und Seriennummern.

Die Funktion VER$ gibt einen String mit sechs Zeichen zurück, der die Version des Betriebssystems, mit dem Sie arbeiten, angibt. Sie erhalten diesen String, wenn Sie im EDIT-Modus ver$ RTN eingeben, in der Anzeige.

Temperaturbereiche

Beachten Sie die folgenden Temperatur- und Luftfeuchtigkeitsgrenzen des HP-75.

Betriebstemperatur: 0° bis 45° C.

Ladetemperatur: 10° bis 40° C.

Lagertemperatur: –40° bis 55° C.

Luftfeuchtigkeit bei Lagerung und Betrieb: 0 bis 95% relative Feuchte.

Ihr Computer sollte niemals außerhalb dieser Grenzen gelagert oder betrieben werden. Hohe Temperaturen sind für den Computer und die Batterien äußerst schädlich. Temperaturschwankungen von einem Extrem ins andere erzeugen Spannungen und verringern die Zuverlässigkeit des Computers. Sie erhalten die größtmögliche Zuverlässigkeit (d.h. minimale Ausfallsrate) bei normaler Zimmertemperatur.

Warnungen

Bestimmte elektronische Schaltkreise des HP-75 arbeiten kontinuierlich und sind daher potentiell anfällig für Unterbrechung oder Beschädigung. Unterbrechungen oder Schäden können entstehen, wenn der Batteriesatz oder ein Einsteckmodul ein- oder ausgebaut wird, solange das Ladegerät angeschlossen ist, wenn das Gerät elektrostatischen Entladungen oder starken Magnetfeldern ausgesetzt wird, oder wenn Peripheriegeräte angeschlossen werden, die von Hewlett-Packard nicht für die Benutzung mit dem HP-75 empfohlen werden. Beachten Sie die folgenden Warnungen.

WARNUNG

- Entfernen Sie immer das Adapter/Ladegerät, wenn Sie den Batteriesatz oder ein Einschubmodul ein- oder ausbauen.
- Fassen Sie den Computer an, wenn Sie ihn auf den Einbau des Batteriesatzes oder eines Einsteckmoduls vorbereiten. Besonders guten Schutz vor elektrostatischen Entladungen können Sie erreichen, wenn Sie die Mulde am Boden des Computers berühren.
- Berühren Sie keinen der Einschubschächte mit den Händen, Werkzeug oder anderen Fremdkörpern.
- Schalten Sie das Gerät aus, bevor Sie den Batteriesatz oder ein Einsteckmodul ein- oder ausbauen.

Genauigkeit der Uhr

Die Systemuhr wird mit einem Quarzkristall auf eine Abweichung von maximal 3 Minuten pro Monat unter den ungünstigsten Temperaturbedingungen geregelt. Typisch ist eine Ungenauigkeit von 1 1/2 Minuten pro Monat. Der Justiervorgang, der in Abschnitt 6 (Seite 95) beschrieben wird, macht Ganggenauigkeiten von 15 Sekunden pro Monat möglich. Der Uhrenquarz leidet unter Temperaturschwankungen, Schlageinwirkungen, Feuchtigkeit und Alterungserscheinungen. Die beste Ganggenauigkeit erreichen Sie bei 25° C ± 5° C. Bei extremem Wechsel der Betriebsbedingungen muß die Uhr gegebenenfalls neu justiert werden.

Abweichungen des BASIC-Interpreters vom ANSI Standard

Der BASIC Sprachinterpreter des HP-75 entspricht der ANSI (American National Standards Institute) Norm für Minimal-BASIC mit den nachfolgenden Ausnahmen. Die Übereinstimmung mit der Norm wurde vom NBS (National Bureau of Standards) in einer Testreihe mit dem HP-75 Interpreter festgestellt. Die Dokumentation dieser Testreihe ist beim National Bureau of Standards, U.S. Department of Commerce, Washington D.C., 20234 als NBS Special Publications 500-70/1 und 500-70/2 erhältlich.

HP-75 Erweiterungen gegenüber dem Minimal-BASIC

Der HP-75 erweitert das Minimal-BASIC in den folgenden Punkten (die Zahlen in Klammern beziehen sich auf die Programmnummern in der Testreihe):

- Einfache numerische Variablen und numerische Feldvariablen können den gleichen Namen besitzen. Der HP-75 unterscheidet sie durch Anwesenheit oder Abwesenheit von Klammern. (Nr.79)

  Beispiel: Die Variablen `A` und `A()` werden als verschieden betrachtet.

- Variablen und Strings werden auf «undefiniert» initialisiert; wenn sie vor der Wertzuweisung abgerufen werden, ergibt dies Warnung 7 – `no value`, und numerische Variablen erhalten den Wert 0, Stringvariablen erhalten den Wert `" "` (Nullstring) zugewiesen. (Nr.23)
- Der Zeichensatz wurde auf den kompletten ASCII-Zeichensatz von 0 bis 255 erweitert. (Nr.93.1, Nr.102, Nr.112)
- Der HP-75 akzeptiert einfache und doppelte Anführungszeichen in einem Literal, wenn diese nicht die ersten Zeichen der Eingabe sind. (Nr.112)
- Bei der Programmeingabe nimmt der HP-75 Leerzeichen am Zeilenanfang und in Schlüsselworten (nach dem zweiten Zeichen) und fehlende Leerzeichen zwischen Schlüsselworten an. Nach der Eingabe der Zeile entfernt der Interpreter überzählige Leerstellen und fügt Leerzeichen ein, wo es die Lesbarkeit erfordert. (Nr.187, Nr.190, Nr.191)
- Das Schlüsselwort `LET` ist bei einer Wertzuweisung optional. (Nr.185)
- Der Interpreter setzt ein unsichtbares `END` ans Ende jedes BASIC- Files; ein Programm, das bei seiner letzten Zeile endet, braucht also die Anweisung `END` nicht zu enthalten. Der Interpreter gestattet auch, die Anweisung `END` mehr als einmal in einem Programm zu benutzen. (Nr.3, Nr.4)
- Da der HP-75 Variablen und Funktionen vor der Ausführung eines Programms vorinitialisiert, müssen benutzerdefinierte Funktionen nicht in Zeilen mit niedriger Nummer definiert als aufgerufen werden. (Nr.157, Nr.159, Nr.162)
- Der HP-75 gestattet `DATA` Anweisungen ohne Datenelemente. (Nr.105)
- Das System erzeugt keinen Fehler, wenn ein Programm endet, bevor eine `FOR` Schleife mit `NEXT` abgeschlossen wurde. (Nr.50)

Abweichungen des HP-75 gegenüber Minimal-BASIC

Der HP-75 stimmt in den folgenden Punkten nicht mit Minimal-BASIC überein:

- Beim Eintritt in eine Schleife weist der HP-75 den Variablen nach `FOR` ihre Werte von links nach rechts zu.

  Beispiel: Die folgende Schleife wird vom HP-75 nur einmal durchlaufen (nach ANSI müsste sie sechsmal durchlaufen werden):

```
J=-2
FOR J=9 TO J STEP J
:
NEXT J
```

  ANSI verlangt, daß Endwert und Inkrement nur einmal beim Eintritt in die Schleife bestimmt werden. Dies erfüllt der HP-75, aber erst *nach* der Spezifikation des Anfangswerts.

- Folgende Tabelle enthält die Reaktionen des HP-75 und die ANSI- Anforderungen bei Eingabefehlern:

Eingabe	HP-75		ANSI
	Default On	Default Off	
Zahlen-Underflow	WARNUNG System gibt 0 vor	FEHLER Neueingabe erforderlich	System gibt 0 vor
Zahlen-Overflow	WARNUNG System gibt `INF` mit Vorzeichen vor	FEHLER Neueingabe erforderlich	Neueingabe erforderlich
Zuweisung	Erfolgt nach Überprüfung jedes einzelnen Datenelements.		Erfolgt nach Überprüfung aller Daten.

  Wenn ein Programm die folgende Anweisung enthält:

```
INPUT I, A(I), X
```

  und die Eingabe lautet:

```
1,2,'abc'
```

  dann ist `'abc'` eine unzulässige Eingabe, und nach ANSI dürfen die Werte für `I` und `A(I)` nicht zugewiesen werden, bevor die Eingabe vollständig richtig ist. Der HP-75 verlangt eine neue Eingabe aller Daten, aber `I` und `A(I)` erhalten ihre neuen Werte `1` und `2` schon zuvor. Wenn Sie nun `2,3,4` eingeben, erhält `A(1)` den Wert `2` und `A(2)` den Wert `3`. Nach ANSI wäre `A(2)` ebenfalls `3`, aber `A(1)` wäre undefiniert. (Nr.108.3)

Stromversorgung

Stromverbrauch

Der HP-75 verbraucht am wenigsten Strom bei ausgeschalteter Anzeige (nach [SHIFT] [ATTN], `BYE` oder der 5minütigen Ausschaltfrist). Bei eingeschaltetem Computer wird mehr Strom verbraucht, noch mehr bei der Ausführung von Programmen, und zur Erzeugung eines Tonsignals benötigt er das Vierfache seines Stromverbrauchs im Bereitschaftszustand. Weiterhin erhöht die Anweisung `standby on` den Stromverbrauch im Bereitschaftszustand auf den bei der Ausführung eines Programms. Wenn der HP-75 nicht vom Netz betrieben wird, kann er mit einem voll aufgeladenen Batteriesatz mehr als 20 Stunden bei Zimmertemperatur (etwa 25° C) arbeiten.

Schalten Sie den HP-75 aus (mit [SHIFT] [ATTN]), bevor Sie ihn an die Steckdose anschließen, damit der Speicherinhalt nicht durch unerwartete Spannungsspitzen verändert wird. Beim Betrieb vom Netz benutzt der HP-75 seinen Batteriesatz als Hilfsstromversorgung, ohne im Normalfall Strom daraus zu entnehmen.

Sie können den HP-75 nicht beschädigen, wenn Sie ihn ohne Batterien betreiben, *aber* Sie können im Falle eines Stromausfalls oder einer Stromunterbrechung den gesamten Speicherinhalt verlieren.

Spannungsüberwachung

Der HP-75 besitzt zwei Spannungskontrollfunktionen, um Ihre Speicherinhalte zu schützen.

Nach dem ersten Hinweis auf den Abfall der Versorgungsspannung sollten Sie den Batteriesatz so bald wie möglich aufladen oder ersetzen.

- Sobald die Batteriespannung unter einen vorgegebenen Wert abfällt, wird die Statusanzeige **BATT** gesetzt; danach haben Sie je nach Zustand der Batterien noch 5 Minuten bis zu 2 Stunden Betriebszeit zur Verfügung.
- Wenn die Betriebsspannung nach dieser Zeitperiode noch weiter abfällt, beendet der HP-75 seinen Normalbetrieb (Programmausführung, Tonsignale, Anzeigeausgaben, Magnetkartenleseroperationen und HP-IL Operationen). Der Batteriesatz versorgt jedoch weiterhin die Uhr und die RAM-Schaltkreise mit Strom. Wenn der HP-75 durch einen fälligen Termin oder mit der Taste [ATTN] wieder eingeschaltet wird, wird kurz die folgende Meldung angezeigt:

```
WARNING: low batteries
    BATT
```

  Sofort danach schaltet sich der HP-75 wieder aus. Drücken Sie jetzt nicht mehr die Taste [ATTN], bevor Sie den Batteriesatz ersetzt oder den HP-75 ans Netz angeschlossen haben. Andernfalls wird der Computer vollständig zurückgesetzt. Die Uhr und der Speicher können noch einige Tage weiter arbeiten, wenn nicht versucht wird, den Rechner in diesem Zustand einzuschalten.

Wenn sich der Batteriesatz der vollständigen Entladung nähert, setzt sich der HP-75 selbst zurück und verliert den gesamten Speicherinhalt. Der HP-75 setzt sich auch dann selbst zurück, wenn Sie bei schwachen Batterien versuchen, den Computer wieder einzuschalten (mit [ATTN]), oder einen Termin zu bestätigen.

Aufladen der Batterien

Der serienmäßige Nickel-Cadmium Batteriesatz wird immer aufgeladen, wenn der HP-75 mit einem Netzadapter/Ladegerät an das Netz angeschlossen ist. Das vollständige Aufladen nach totaler Entladung dauert etwa acht Stunden. Bei kürzeren Aufladezeiten wird die zu erwartende Betriebszeit entsprechend verkürzt. Der Batteriesatz kann niemals überladen werden. Wenn Sie den Batteriesatz jedoch außerhalb des Ladetemperaturbereichs (Seite 267) aufladen, kann die Lebensdauer ernsthaft vermindert werden.

Wenn Sie den HP-75 über lange Zeit ununterbrochen vom Netz betreiben, sind die Nickel-Cadmium Batterien nach einiger Zeit nicht mehr auf ihre volle Kapazität aufladbar. Es ist deshalb ratsam, das Netzadapter/Ladegerät alle paar Monate zu entfernen und den Akkumulatorensatz bei normaler Benutzung zu entladen, bis die Statusanzeige **BATT** gesetzt wird. Wenn danach das Netzadapter/Ladegerät angeschlossen wird, werden die Nickel-Cadmium Batterien wieder auf ihre volle Kapazität aufgeladen.

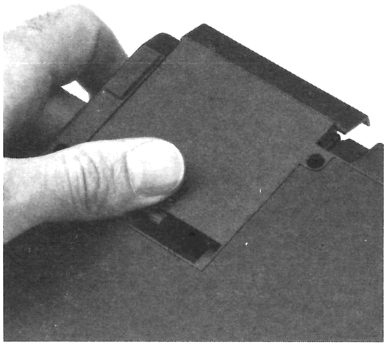
Ersetzen des Batteriesatzes

Wenn der Batteriesatz ersetzt werden soll, müssen Sie zuerst das Netzadapter/Ladegerät entfernen. Nach der Entnahme des Batteriesatzes sind die Uhr und die RAM-Kreise für mindestens 30 Sekunden gegen Informationsverlust geschützt. Da die Schutzzeit gegen den Verlust des Speicherinhalts begrenzt ist, sollten Sie die folgende Anleitung durchlesen, bevor Sie mit dem Ersetzen des Batteriesatzes beginnen. Beachten Sie die Warnungen an früherer Stelle in diesem Anhang (Seite 267). Folgende weitere Vorsichtsmaßnahmen sollten Sie einhalten, wenn Sie den Batteriesatz entfernen:

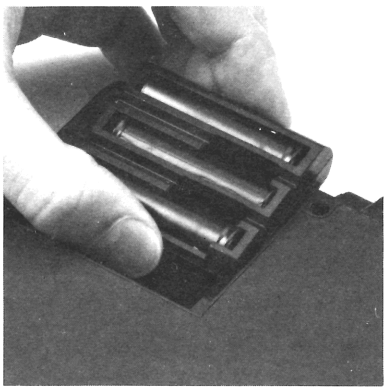
- Drücken Sie nicht die Taste [ATTN], wenn keine Batterien im Batteriefach sind. Der HP-75 setzt sich zurück, wenn er ohne entsprechende Stromquelle eingeschaltet wird.
- Es ist empfehlenswert, wichtige Files auf ein Massenspeichermedium zu kopieren, bevor Sie den Batteriesatz entfernen, um im Fall eines Zurücksetzens den Verlust kritischer Information zu vermeiden.

So wird der Batteriesatz ersetzt:

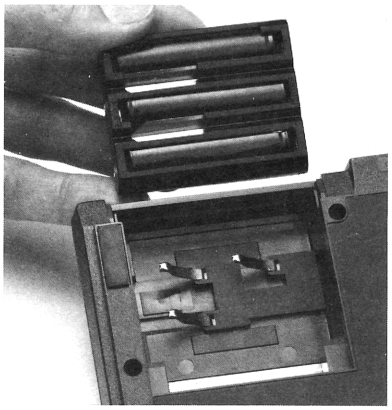
1. Geben Sie im EDIT-Modus `alarm off` [RTN] ein, um ein Fälligwerden von Terminen zu verhindern, solange der Computer batterielos ist.
2. *Schalten Sie den HP-75 aus.* Drücken Sie dazu [SHIFT] [ATTN] oder geben Sie `bye` [RTN] ein.
3. *Entfernen Sie das Netzadapter/Ladegerät vom HP-75.*
4. Drehen Sie den Computer um, und legen Sie ihn auf eine flache, glatte Oberfläche.
5. Drücken Sie die Lasche am Deckel des Batteriefachs nach unten, und schieben Sie diesen in Richtung Rückseite des Computers.



6. Entfernen Sie den Batteriesatz, indem Sie ihn aus dem Batteriefach heben.



7. Richten Sie die Kontakte des Ersatz-Batteriesatzes auf die richtigen Federkontakte aus, und setzen Sie ihn in das Batteriefach ein.



8. Drücken Sie den Batteriesatz nach unten, und schieben Sie den Deckel des Batteriefachs in den Führungsschienen wieder zurück, bis er einrastet.

Der Nickel-Cadmium Batteriesatz benutzt die beiden parallelen Kontakte im Batteriefach. Durch falsches Einsetzen des Batteriesatzes wird der Computer nicht beschädigt, er setzt sich aber zurück, wenn er in diesem Zustand eingeschaltet wird.

Pflege des Batteriesatzes

Wenn die Rückseite des HP-75 handwarm ist, wird der Batteriesatz aufgeladen. Nach acht oder mehr Stunden Aufladezeit sollte der Batteriesatz voll geladen sein. Wenn sich ein aufladbarer Batteriesatz sehr schnell zu entladen scheint, kann dies daran liegen, daß der Batteriesatz wiederholt bei geringfügigem Spannungsunterschied geladen worden ist, da der HP-75 nur wenig Strom entnommen hat. Entfernen Sie das Netzadapter/Ladegerät, geben Sie im EDIT-Modus `standby on` RTN ein und lassen Sie den HP-75 eingeschaltet, bis die Betriebsspannung soweit gefallen ist, daß der HP-75 den Betrieb aufgibt und sich selbst ausschaltet (wie unter «Spannungsüberwachung» beschrieben). Schließen Sie das Gerät dann für mindestens zehn Stunden ans Netz an, um den Batteriesatz vollständig aufzuladen.

WARNUNG

Der Batteriesatz darf nicht zerstört oder verbrannt werden – er könnte platzen oder Giftstoffe an die Umwelt abgeben.

Die Batterieanschlüsse dürfen nicht verbunden oder auf andere Weise kurzgeschlossen werden – der Batteriesatz könnte schmelzen oder Brände verursachen.

Der Batteriesatz unterliegt einer Gewährleistungsfrist von 1 Jahr und wird ersetzt, wenn er in dieser Zeit nicht einwandfrei arbeitet. Geben Sie einen defekten Batteriesatz entsprechend den Versandhinweisen auf Seite 282 an Hewlett-Packard zurück. (Wenn Ihnen die Ursache des Ausfalls nicht bekannt ist, sollten Sie den HP-75 vollständig mit seinem Batteriesatz und dem Netzadapter/Ladegerät einsenden.) Wenn für den Batteriesatz keine Gewährleistungspflicht mehr besteht, sollten Sie bei Ihrem HP-Händler Ersatz anfordern.

Pflege von Kartenleser und Magnetkarten

Reinigen von Magnetkarten

Ein einwandfreier Betrieb des Magnetkartenlesers erfordert die Reinhaltung der Magnetkarten. Die Oberflächen der Magnetkarten können Staub und Öl ansammeln und damit den Informationsaustausch mit dem HP-75 empfindlich stören. Ihre Finger stellen eine häufige Quelle von Schmutz und Öl dar. Fassen Sie Ihre Magnetkarten nur an den Kanten an. Sie können die Magnetkarten mit einem weichen, sauberen, fusselfreien Tuch reinigen, das mit Isopropylalkohol befeuchtet wurde. Legen Sie die Magnetkarte auf eine glatte, saubere Fläche und wischen Sie mit dem Reinigungstuch kräftig über die magnetisierte Oberfläche (die nicht beschriftete Seite) der Magnetkarte.

Knicken, Biegen oder Zerkratzen der Magnetkarte kann irreparable Beschädigungen hinterlassen. Es ist zu empfehlen, Sicherungskopien von wichtigen Files in einem Magnetkartenhalter aufzubewahren, für den Fall, daß die Originalkarte eines Files beschädigt wird.

Reinigen des Lesekopfes

Der Kartenleserkopf ist vergleichbar mit dem Tonkopf eines High-Fidelty Aufnahmegerätes. Auch hier kann eine Ansammlung von Schmutz und anderen Fremdkörpern auf dem Kopf den Kontakt zwischen Kopf und Magnetkarte beeinträchtigen. Wenn Sie schmutzige Magnetkarten durch den Magnetkartenleser führen, wird dessen Betriebszuverlässigkeit vermindert. Sie können den Lesekopf mit der Reinigungskarte (die dem Benutzerpaket beiliegt) säubern, indem Sie diese in Pfeilrichtung (wie bei einem Lesevorgang) ein- oder zweimal durch den Magnetkartenleser ziehen, wenn Sie vermuten, daß der Magnetkartenleser nicht einwandfrei arbeitet. Normalerweise sollte die Reinigung mit der Reinigungskarte nur einige wenige Male während der Lebensdauer des Computers erforderlich sein.

VORSICHT

Häufige Benutzung der Reinigungskarte kann eine starke Abnutzung des Lesekopfes verursachen.

Beschriften von Magnetkarten

Sie können die Vorderseite einer Magnetkarte beschriften, wenn Sie darauf achten, daß die Oberfläche der Magnetkarte dabei nicht eingedrückt wird. Tintenfilzstifte (verwenden Sie keine gewöhnlichen Filzstifte auf Wasserbasis), Tuschezeichner oder technische Zeichenstifte mit Tinte sowie Bleistifte sind gut zur Beschriftung der Magnetkarten geeignet. Die meisten Tinten benötigen einige Sekunden zum Trocknen. Bleistift kann verschmieren, ist aber ausradierbar.

Allgemeine Reinigungshinweise

Der HP-75 kann mit einem weichen Tuch gereinigt werden, das mit sauberem Wasser und gegebenenfalls einem milden Reinigungsmittel angefeuchtet ist. Das Tuch darf nicht triefend naß sein, und es sollte kein Wasser in das Gerät hineinlaufen. Vermeiden Sie Scheuermittel, insbesondere auf dem Anzeigefenster.

Es ist empfehlenswert, mit [SHIFT] [ATTN] das Tastenfeld zu desaktivieren, bevor Sie den HP-75 reinigen.

Einschubmodule

Ihr HP-75 verfügt über einen im Gehäuse befindlichen Anschluß für ein optionales Speichermodul und drei äußere Einschubschächte für Einschub-ROMs, mit denen Sie die Fähigkeiten Ihres Computers erweitern können. Vor dem Versand wird in jeden der drei äußeren Einschubschächte eine Modul-Attrappe eingesetzt, um die dazugehörenden Schaltkreise zu schützen. Diese drei Einschubschächte sollten verschlossen bleiben, wenn sie nicht benutzt werden, damit keine Fremdstoffe in den Computer eindringen können.

Hinweis: Schalten Sie den HP-75 auf jeden Fall aus (drücken Sie [SHIFT] [ATTN]), bevor Sie ein Modul ein- oder ausbauen. Andernfalls kann der Computer zurückgesetzt werden, und alle gespeicherte Information geht damit verloren.

VORSICHT

Beachten Sie die auf Seite 267 gelisteten Vorsichtsmaßnahmen.

Wenn ein Modul beim Einschieben in einen Schacht klemmt, ist es möglicherweise verkehrt herum eingesetzt. Wenn Sie es dennoch weiter einschieben, können Sie das Modul oder den Computer damit beschädigen.

Sie finden Anleitungen zur Installation und Benutzung des optionalen Speichermoduls und der Einschubmodule in der jedem Zubehörteil beiliegenden Dokumentation.

Spezielle Files

Drei spezielle Filetypen – Austausch-, LEX und ROM Files – stehen Ihnen über den TRANSFORM Befehl, vorbespielte Magnetkarten und Bandkassetten bzw. über Einschub-ROMs zur Verfügung.

Austauschfiles (TRANSFORM, LIF1)

Austauschfiles oder LIF1 (*Logical Interchange Format*) Files dienen zum Austausch von Informationen zwischen dem HP-75 und anderen Computern. Jeder Textfile und jeder nichtprivate BASIC-File im Speicher kann mit Hilfe des Befehls TRANSFORM in einen Austauschfile umgewandelt werden und dann auf ein Massenspeichermedium abgelegt werden, das von einem anderen Computer gelesen werden kann. Entsprechend kann ein Austauschfile auch von einem Massenspeichermedium in den Speicher eingelesen und dann in einen BASIC- oder Textfile transformiert werden. Austauschfiles sind wie Textfiles aus alphanumerischer Information zusammengesetzt.

Folgende Operationen können mit Austauschfiles durchgeführt werden:

- Katalogisierung (CAT). Das Symbol I bezeichnet einen Austauschfile im Systemkatalog.
- Kopieren von und auf Massenspeicher (COPY).
- Umwandlung in BASIC- oder Textfiles (TRANSFORM).
- Namensänderung (RENAME), Löschen (PURGE) und Duplizieren im Speicher (COPY).

Folgende Operationen können mit Austauschfiles *nicht* durchgeführt werden:

- Ausführung als Programm (RUN, CALL).
- Editieren (EDIT).
- Listen (LIST, PLIST).

Wenn Sie versuchen, einen Austauschfile auszuführen, zu editieren oder auszulisten, erhalten Sie Fehler 68 – `wrong filetype`.

Mit dem Befehl `TRANSFORM` können Sie einen Filetyp im Speicher (BASIC-, Text- oder Austauschfile) in einen anderen Filetyp umwandeln.

```
TRANSFORM ['Filename'] INTO BASIC
                           TEXT
                           LIF1
```

Befindet sich zum Beispiel der BASIC-File `'FINANZ'` im Speicher, dann wird dieser File mit der Eingabe `transform 'finanz' into text` RTN in einen Textfile umgewandelt. Der Katalogeintrag des Files zeigt ein `T` für Text. Ein Listing des umgewandelten Textfiles sieht zwar nicht anders aus als ein Listing des ursprünglichen BASIC-Files, diese Files sind jedoch verschieden. Textfiles werden intern als Folgen von Zeilennummern und ASCII-Zeichen abgespeichert, während BASIC-Files intern als eine Folge von Instruktionen im Maschinencode abgelegt werden. Die umgekehrte Operation, `transform 'finanz' into basic`, wandelt den File in einen ausführbaren BASIC-File um.

BASIC- und Textfiles können in Austauschfiles umgewandelt werden, wie auch Austauschfiles in BASIC- oder Textfiles umgewandelt werden können.

Der Befehl `TRANSFORM` hat folgende Auswirkungen:

Quell-file	TRANSFORM INTO BASIC	TRANSFORM INTO Text	TRANSFORM INTO LIF1
BASIC	Keine Aktion.	Maschinencode wird in Textzeilen umgewandelt.	Maschinencode wird in Textzeilen und dann Zeile für Zeile in Zeicheninformation umgewandelt.
Text	Der File wird Zeile für Zeile in Maschinencode umgewandelt.	Keine Aktion.	Der File wird Zeile für Zeile in Zeicheninformation umgewandelt.
LIF1	Der File wird Zeichen für Zeichen in Textzeilen und dann Zeile für Zeile in Maschinencode umgewandelt.	Der File wird Zeichen für Zeichen in Textzeilen umgewandelt.	Keine Aktion.

Die Umwandlung eines Files kann, je nach Länge des Files, mehrere Sekunden in Anspruch nehmen. Während der Umformung wird das gesamte Tastenfeld einschließlich der Taste ATTN desaktiviert, um eine Mischung von Filetypen zu verhindern. Der Inhalt der Anzeige zu Beginn der Ausführung des `TRANSFORM` Befehls bleibt während der gesamten Transformation in der Anzeige stehen. Nach beendeter Operation erscheinen die Eingabeaufforderung und der Cursor wieder.

Der Filename im `TRANSFORM` Befehl kann durch einen Stringausdruck, wie zum Beispiel `A$` oder `CAT$(0)`, spezifiziert werden. Wenn kein Filename spezifiziert wird, wandelt der HP-75 den momentanen File um:

- Wenn der momentane File in einen Text oder BASIC-File umgewandelt wird, erscheint danach die Text bzw. BASIC-Eingabeaufforderung, und der Filepointer zeigt auf die erste Zeile des Files.
- Bei der Umwandlung des momentanen Files in LIF1 wird ein neuer Arbeitsfile des derzeitigen Filetyps erzeugt, und der Filepointer zeigt auf Zeile 0 des Arbeitsfiles.
- Wenn der momentane File selbst ein `workfile` ist, sind die Befehle `TRANSFORM INTO TEXT` und `TRANSFORM INTO BASIC` möglich, und der resultierende File ist ein Text- oder BASIC-Arbeitsfile. `TRANSFORM INTO LIF1` ist jedoch nicht erlaubt und erzeugt Fehler 63 – `invalid filespec`; geben Sie dem File in diesem Fall zuerst einen Namen und führen Sie den Befehl `TRANSFORM` erneut aus.

Wenn ein Text- oder Austauschfile in einen BASIC-File umgeformt wird, muß jede Zeile des umgeformten Files eine sinnvolle Programmanweisung sein. Wenn der HP-75 bei der Umformung in BASIC eine unverständliche Zeichenfolge antrifft, wird die gesamte Zeile, die diese Zeichen enthält, in eine *Kommentarzeile* umgewandelt. Der HP-75 beginnt die Zeile nach der Zeilennummer mit einem Ausrufezeichen und nachfolgendem Fragezeichen (! ?).

Beispiel:

```
110 A=A MOD 256
```

Diese Zeichenfolge kann nicht als gültige BASIC-Anweisung interpretiert werden...

```
110 ! ? A=A MOD 256
```

...deshalb wird die gesamte Zeile in eine Kommentarzeile umgewandelt.

Wenn der resultierende BASIC-File eine oder mehrere Kommentarzeilen dieser Form enthält, meldet der HP-75 nach beendeter Umformung Fehler 78 – syntax. Wenn Sie den transformierten BASIC-File editieren, können Sie mit einem speziellen FETCH-Befehl jede Zeile auffinden, die der HP-75 nicht interpretieren konnte.

Beispiel:

```
>FETCH '! ?', 0 ■
```

Sucht von Zeile 0 des Files an nach dem nächsten Auftreten von ! ?.

```
110 ! ? A=A MOD 256
```

Diese Zeile kann nun korrigiert und als BASIC-Anweisung eingegeben werden.

Wenn Sie andererseits einen BASIC-File in einen Text- oder LIF1-File umwandeln, werden Zeilen des Quellprogramms, die mit ! ? (mit oder ohne dazwischenstehendem Leerzeichen) beginnen, im Text- oder LIF1-File *nicht* in der Kommentarform abgespeichert. Sie sollten also vermeiden, ein Fragezeichen als erstes Zeichen einer Kommentarzeile zu wählen, damit die Kommentarzeichen ! ? bei einer Transformation in Text oder LIF1 nicht entfernt werden.

Wenn bei einer Umformung von BASIC in Text eine Zeile die Länge von 94 Zeichen (einschließlich Zeilennummer) überschreitet; werden alle Zeichen nach Zeichen 94 in der BASIC-Quellzeile gelöscht, es wird Warnung 67 – line too long – angezeigt, und die Umformung wird fortgesetzt. Die resultierende Textzeile enthält ein Fragezeichen an der ersten Stelle nach der Zeilennummer, gefolgt von den ersten 94 Zeichen der BASIC-Quellzeile. Diese Situation kann eintreten, wenn Sie zum Beispiel einen BASIC-File mit langen DATA Anweisungen in Text umwandeln.

Bei einer Umwandlung von Text in BASIC ist es möglich, daß eine oder mehrere der entstehenden Programmzeilen bei der Auflistung mehr als 94 Zeichen enthalten; dies kann zum Beispiel der Fall sein, wenn eine Zeile im Quelltext zahlreiche Abkürzungen enthält.

Beispiel:

```
:10■op.a.d.@op.a.d.@op.a.d.@op.a
.d.@op.a.d.
```

Bei der Umformung in BASIC ergibt diese Textzeile fünfmal den Befehl OPTION ANGLE DEGREES in einer langen Zeile.

Während der Umformung wird kein Fehler gemeldet, und alle Instruktionen in der Zeile werden im BASIC-File gespeichert. Beim Versuch, die sich ergebende lange Zeile zu listen oder zu editieren, erhalten Sie Warnung 67 – line too long; nur die ersten 94 Zeichen der Zeile werden angezeigt und stehen zur Editierung zur Verfügung. Wenn der File nun in Text zurückverwandelt wird, wird die lange Zeile nach dem 94. Zeichen abgeschnitten.

Auch der File keys kann umgeformt werden. Geben Sie dazu dem File zuerst einen anderen Namen, und verwenden Sie den neuen Namen im Befehl TRANSFORM.

Von anderen Computern als dem HP-75 erzeugte Austauschfiles können längere Zeilen, als der HP-75 gestattet, oder unnumerierte Zeilen enthalten. Solche Files können zwar in den Speicher eingelesen, nicht aber in Text oder BASIC umgewandelt werden. Ein Austauschfile mit überlangen Zeilen erzeugt Fehler 67 – `line too long`; ein Austauschfile ohne Zeilennummern erzeugt Fehler 87 – `bad expression`. In beiden Fällen wird die Umformung nicht durchgeführt; der LIF1-File bleibt ein LIF1-File.

Wenn sich das derzeit laufende Programm selbst umformt, wird die Ausführung unterbrochen, das Programm deallokatisiert und dann umgewandelt.

Zur Ausführung des `TRANSFORM` Befehls wird einiger Systemspeicherplatz benötigt; deshalb können einige verschachtelte zulässige Ausdrücke bei der Umformung von Text- oder LIF1-Files in BASIC in Kommentarzeilen des Typs `! ?` umgewandelt werden.

LEX-Files (`COPY`)

LEX-Files oder *Language Extension Files* sind speziell verschlüsselte Programme, die die Fähigkeiten des HP-75 erweitern. Ein LEX-File kann auf einem Massenspeichermedium oder in einem Einsteck-ROM abgelegt sein. LEX-Files werden von Massenspeichermedien genau wie andere Files mit `COPY` in den Speicher (RAM) eingelesen. Ein LEX- File im RAM wird im Systemkatalog mit dem Symbol `L` gekennzeichnet. Ein LEX-File im ROM erscheint nicht im Systemkatalog, arbeitet aber genauso wie ein LEX-File im RAM.

Mit Hilfe von LEX-Files können neue Schlüsselworte für den Benutzer verfügbar gemacht werden; dazu gehören zusätzlich Funktionen, Anweisungen und Befehle, die nicht standardmäßig zum Betriebssystem des HP-75 gehören. LEX-Files können nicht gelistet, editiert oder ausgeführt werden; LEX-Files im RAM können aber dupliziert, umbenannt und gelöscht werden.

Genaue Anleitungen zur Benutzung von LEX-Files liegen jeweils dem Speichermedium mit dem LEX-File bei. Bei der Verwendung von Programmen mit Schlüsselworten, die im LEX-File definiert werden, muß der LEX- File immer (entweder im RAM oder ROM) vorhanden sein. Falls der LEX- File nicht vorhanden ist, meldet das System bei der Ausführung des Programmes Fehler 18 – `ROM missing`. Auch beim Editieren eines Programmes im nicht initialisierten Zustand kann in diesem Fall nicht auf Zeilen mit LEX Schlüsselworten zugegriffen werden. Wenn das Programm initialisiert ist, werden Zeilen mit LEX-Schlüsselworten durch Zeilen mit `ROM missing` ersetzt (sofern die jeweilige Operation das Programm deallokatisiert).

ROM Files (`RUN`, `LIST`, `COPY`, `MERGE`)

ROM-Files sind permanent in Einsteck-ROMs (Read-Only-Memory), die als optionales Zubehör zum HP-75 erhältlich sind, abgelegt. ROM-Files können von jedem der Filetypen BASIC, Text und LEX sein. BASIC ROM-Files können gestartet (`RUN`) und aufgerufen (`CALL`) werden; LEX ROM-Files stellen dem System neue Schlüsselworte zur Verfügung, sobald das ROM-Modul in den HP-75 eingesteckt ist.

ROM-Files können auf keine Weise katalogisiert, editiert, gelistet, gelöscht, umbenannt oder geändert werden; es ist auch nicht möglich, mit `READ #` einen ROM-File zu lesen. Einzelne nichtprivate ROM-Files können jedoch mit dem Befehl `COPY` in den Speicher eingelesen werden.

```
COPY 'ROM Filename' TO 'RAM Filename'
```

Weiterhin kann ein BASIC- oder Text-ROM File vollständig oder teilweise mit dem Befehl `MERGE` in den momentanen File im Speicher eingefügt werden.

```
MERGE 'ROM Filename' [, Anfangszeilennummer [, Endzeilennummer]]
```

Sobald er im RAM steht, kann der File katalogisiert, editiert, gelistet, umbenannt usw. werden. Wenn Sie einen ROM-File auf ein Massenspeichermedium kopieren wollen, müssen Sie den File zuerst in den Speicher kopieren. Beachten Sie, daß der HP-75 zuerst auf den File im RAM zugreift, wenn sich zwei Files mit gleichem Namen im ROM und im RAM befinden.

Detailliertere Betriebshinweise finden Sie in der dem ROM beiliegenden Dokumentation.

Andere Files (?)

Ein Fragezeichen, ?, als Symbol für den Filetyp auf einem Massenspeichermedium zeigt an, daß es dem HP-75 nicht möglich ist, die in diesem File enthaltene Information zu nutzen. Ein File dieses Typs kann sich ergeben, wenn ein anderer Rechner oder Computer Information auf diese Massenspeichereinheit geschrieben hat.

Beim Versuch, einen ? File in den Speicher des HP-75 einzulesen, entsteht Fehler 68 – `wrong file type`. Unbekannte Files können umbenannt (`RENAME`), dupliziert (`COPY`) und vom Massenspeicher gelöscht (`PURGE`) werden.

Funktionsprüfung

Wenn Sie vermuten, daß Ihr HP-75 nicht mehr einwandfrei funktioniert und eventuell eine Reparatur erforderlich ist, sollten Sie den folgenden Funktionstest ausführen:

1. Schalten Sie das Gerät mit [SHIFT] [ATTN] aus, und entfernen Sie alle HP-IL Einheiten.
2. Schließen Sie den Computer mit dem Adapter/Ladegerät ans Netz an.
3. Schalten Sie das Gerät mit [ATTN] wieder ein. Sie sollten eine dieser Anzeigen erhalten:

```
>
```

Die BASIC-Eingabeaufforderung und der Cursor.

```
:
```

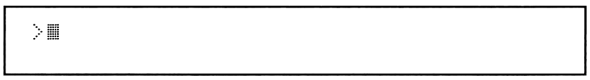
Die Text-Eingabeaufforderung und der Cursor.

4. Führen Sie die Funktion `pi` [RTN] aus. Wenn das Ergebnis `3.14159265359` richtig angezeigt wird, arbeiten etwa 60 Prozent der Schaltkreise einwandfrei.
5. Wenn der Computer eine bestimmte Operation, wie zum Beispiel das Kopieren eines Files auf Magnetkarte, wiederholt nicht durchführt, oder wenn er wiederholt eine Fehlermeldung wie zum Beispiel `bad parameter` anzeigt, sollten Sie die Anleitungen zu dieser Instruktion in diesem Handbuch nochmals sorgfältig durchlesen; es ist denkbar, daß Sie die Operation nicht richtig spezifizieren.
6. Setzen Sie den Computer zurück, falls die Anzeige leer bleibt, nachdem Sie [ATTN] gedrückt haben:
   a. Entfernen Sie das Netzadapter/Ladegerät.
   b. Entfernen Sie den Batteriesatz.
   c. Drücken Sie [ATTN]. Die Schaltkreise werden sofort entladen.
7. Nach dem Wiedereinbau des Batteriesatzes und dem Anschluß des Netzadapter/Ladegeräts sollte beim Drücken von [ATTN] die Zeitsetzmaske erscheinen:

```
Set Mo/Dy/Year Hr:Mn:Sc AM
```

Geben Sie die Werte für Datum und Zeit ein, und drücken Sie [RTN].

8. Drücken Sie nach dem Setzen der Uhr die Taste [EDIT]:



Die BASIC-Eingabeaufforderung und der Cursor sollten erscheinen. Wiederholen Sie nun Schritt 4.

9. Wenn die Anzeige leer bleibt, oder wenn die Meldung `ERROR: RAM is invalid` angezeigt wird, sollten Sie alle Einsteck-RAMs und -ROMs entfernen, das Gerät zurücksetzen und mit den Schritten 7 und 8 feststellen, ob das Fehlverhalten von defekten oder falsch installierten Modulen erzeugt worden ist.

10. Wenn der HP-75 nicht mehr auf Eingaben reagiert, und die Anzeige «erstarrt», können Sie den Computer mit [SHIFT] [CTL] [CLR] oder durch Entfernen aller Spannungsquellen zurücksetzen, und danach mit den Schritten 7 und 8 auf einwandfreie Funktion prüfen.

11. Sie können die HP-IL auf einwandfreie Funktion prüfen, indem Sie die Buchsen IN und OUT des HP-75 mit einem einzelnen HP-IL Kabel verbinden. Geben Sie danach `assignio` [RTN] ein. Wenn Fehler 56 – `no loop response` – auftritt, arbeitet der HP-IL Schaltkreis fehlerfrei; andere Reaktionen weisen auf ein defektes Kabel oder andere Defekte hin.

Nehmen Sie Kontakt mit Hewlett-Packard auf, wenn Sie die Fehlerursache nicht feststellen können. Die Adressen und Telefonnummern finden Sie unter «Service» in diesem Anhang.

Gewährleistung

Hewlett-Packard gewährleistet, daß Ihr Computer HP-75 frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn das Gerät – direkt oder über einen HP-Vertragshändler – an Hewlett-Packard eingeschickt wird. Die Gewährleistung beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die durch unsachgemäße Veränderungen des Geräts durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt.

Die Gewährleistung gilt nur in Verbindung mit entweder

a) dem von einem Hewlett-Packard Vertragshändler ausgestellten Kaufbeleg und der vollständig ausgefüllten, von diesem Vertragshändler unterschriebenen Servicekarte oder

b) der Originalrechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Gewährleistungsregelung unberührt.

Änderungsverpflichtung

Sämtliche Produkte werden auf der Grundlage der technischen Daten bei der Herstellung verkauft. Hewlett-Packard übernimmt keine Verpflichtung, einmal verkaufte Geräte zu modifizieren oder auf den neuesten technischen Stand zu bringen.

Gewährleistungsinformationen

Wenn Sie bezüglich dieser Gewährleistung Fragen haben, setzen Sie sich bitte mit einem Hewlett-Packard Vertragshändler oder einer Hewlett-Packard Verkaufs- und Serviceniederlassung in Verbindung. Falls Ihnen dies nicht möglich ist, schreiben Sie an:

- In Europa:

Hewlett-Packard S.A.
7,rue du Bois-du-Lan
P.O.Box
CH-1217 Meyrin 2 Genf
Schweiz
Telefon: (022) 83 81 11
Hinweis: Senden Sie keine Computer zur Reparatur an diese Adresse.

- In den U.S.A.:

Hewlett-Packard
Corvallis Division
1000 N.E. Circle Blvd. Corvallis, OR 97330
Telefon: (503) 758-1010

- In anderen Ländern:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, CA 94304 U.S.A.
Telefon: (415) 857-1501
Hinweis: Senden Sie keine Computer zur Reparatur an diese Adresse.

Service

Serviceeinrichtungen

Hewlett-Packard unterhält weltweit Serviceniederlassungen in den meisten wichtigen Ländern. Sie können Ihr Gerät jederzeit von einer Hewlett-Packard Serviceniederlassung reparieren lassen, sei es mit oder ohne Gewährleistung. Nach Ablauf der einjährigen Gewährleistungsfrist werden Ihnen die Reparaturkosten berechnet.

Produkte von Hewlett-Packard werden von jeder Serviceniederlassung normalerweise innerhalb von fünf Arbeitstagen nach Erhalt repariert und wieder ausgeliefert. Dies ist eine Durchschnittszeit, die von der der Auslastung des Servicepersonals abhängen kann. Die Abwesenheitszeit Ihres Gerätes hängt zu einem großen Teil von der Versandzeit ab.

Service in Europa

Die folgende Aufstellung zeigt die Serviceniederlassungen in Europa. Setzen Sie sich in nicht aufgelisteten Ländern mit dem Verkäufer Ihres Computers in Verbindung.

BELGIEN
HEWLETT-PACKARD
BELGIUM SA/NV
Boulevard de la Woluwe 100
Woluwelaan
B-1200 BRÜSSEL
Telefon (2) 762 32 00

DÄNEMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 BIRKERØD (Kopenhagen)
Telefon (02) 81 66 40

DEUTSCHLAND
HEWLETT-PACKARD GmbH
Vertriebszentrale
Berner Straße 117
Postfach 560140
D-6000 FRANKFURT 56
Telefon (0611) 50041

FINNLAND
HEWLETT-PACKARD OY
Revontulentie 7
02100 ESPOO 10 (Helsinki)
Telefon (90) 455 02 11

FRANKREICH
HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de poche
F-91947 Les Ulis Cedex
Telefon (6) 907 78 25

GROSSBRITANNIEN
HEWLETT-PACKARD LTD.
King Street Lane
GB-WINNERSH, WOKINGHAM
BERKSHIRE RG11 5AR
Telefon (734) 784774

ITALIEN
HEWLETT-PACKARD
ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 CERNUSCO SUL NAVIGLIO
(Mailand)
Telefon (2) 90 36 91

NIEDERLANDE
HEWLETT-PACKARD
NEDERLAND B.V.
Van Heuven Goedhartlaan 121
1181 KK AMSTELVEEN (Amsterdam)
P.O. Box 667
Telefon (020) 47 20 21

NORWEGEN
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 OESTERAAS (Oslo)
Telefon (2) 17 11 80

ÖSTERREICH
HEWLETT-PACKARD GmbH
Wagramerstraße-Lieblgasse
A-1220 WIEN
Telefon (222) 23 65 11

OSTEUROPA
Wenden Sie sich bitte an die
Service-Zentrale in Österreich

SCHWEDEN
HEWLETT-PACKARD SVERIGE AB
Box 19
Skalholtsgatan 9, Kista
S-16393 SPANGA/STOCKHOLM
Telefon (8) 750 20 00

SCHWEIZ
HEWLETT-PACKARD (SCHWEIZ) AG
Allmend 2
CH-8967 WIDEN
Telefon (057) 5 01 11

SPANIEN
HEWLETT-PACKARD
ESPAÑOLA S.A.
Calle Jerez 3
E-MADRID 16
Telefon (1) 458 26 00

Service in den U.S.A.

Das Servicezentrum für batteriebetriebene Computerprodukte von Hewlett-Packard in den U.S.A. befindet sich in Corvallis, Oregon:

Hewlett-Packard Company
Corvallis Division Service Department
P.O.Box 999/1000 N.E.Circle Blvd.
Corvallis, OR 97330, U.S.A.
Telefon: (503) 757-2000

Internationale Service-Information

Nicht alle Hewlett-Packard Serviceniederlassungen bieten Reparatur für alle HP Computerprodukte an. Wenn Sie Ihr Gerät bei einem autorisierten Hewlett-Packard Händler gekauft haben, können Sie allerdings sicher sein, daß in dem Land, wo der Computer gekauft wurde, auch Service angeboten wird.

Wenn Sie sich außerhalb des Landes befinden, in dem Sie das Gerät gekauft haben, sollten Sie sich mit der örtlichen Hewlett-Packard Serviceniederlassung in Verbindung setzen und, falls eine Reparatur dort nicht möglich ist, das Gerät an die unter «Service in den U.S.A.» angegebene Adresse schicken. Von dort können Sie auch eine Liste der Serviceniederlassungen in anderen Ländern erhalten.

Sämtliche mit dem Versand verbundenen Kosten gehen zu Ihren Lasten.

Reparaturkosten

Hewlett-Packard erhebt für Reparaturen, die außerhalb der Garantiezeit liegen, Gebühren nach einem festgelegten Satz. In den Reparaturkosten sind Arbeitszeiten und Materialien eingeschlossen. In Europa wird auf den Rechnungsbetrag Mehrwertsteuer erhoben und auf der Rechnung als getrennter Posten ausgewiesen.

Die festgelegten Reparaturkosten gelten nicht für durch Gewalteinwirkung oder Mißbrauch beschädigte Computerprodukte. In solchen Fällen werden die Reparaturkosten auf der Grundlage von Arbeitszeit und Material individuell festgelegt.

Gewährleistung auf Servicearbeiten

Auf alle Reparaturen außerhalb der Garantiezeit wird eine Gewährleistungsgarantie auf Material und Verarbeitung für 90 Tage vom Reparaturdatum an gegeben.

Versandanweisungen

Wenn Ihr Gerät repariert werden muß, senden Sie es bitte mit den folgenden Unterlagen ein:

- eine ausgefüllte Servicekarte, die eine Beschreibung des Problems enthält,
- ein Kassenbeleg oder ein anderes Dokument des Verkaufsdatums, falls die einjährige Gewährleistungsfrist noch nicht abgelaufen ist.

Das Gerät, die Servicekarte, eine kurze Fehlerbeschreibung und gegebenenfalls der Beleg des Kaufdatums sollten in der Originalverpackung oder einer anderen angemessenen Schutzverpackung eingesandt werden, um Versandschäden zu vermeiden. Versandschäden sind in der Jahresgarantie nicht eingeschlossen; Hewlett-Packard empfiehlt, für den Versand zur Serviceniederlassung eine Transportversicherung abzuschließen. Das verpackte Gerät sollte an die nächstliegende von Hewlett-Packard angegebene Sammelstelle oder Serviceniederlassung versandt werden. Lassen Sie sich von Ihrem Händler beraten. (Wenn Sie sich nicht in dem Land aufhalten, in dem Sie das Gerät erworben haben, lesen Sie den vorangegangenen Abschnitt «Internationale Serviceinformation»).

Unabhängig davon, ob für das Gerät noch Gewährleistung besteht oder nicht, gehen die Kosten für den Versand zur Serviceniederlassung zu Ihren Lasten.

Weitere Informationen

Hewlett-Packard bietet keine Serviceverträge an. Ausführung und Design von Computerprodukten sind Eigentum von Hewlett-Packard; Servicehandbücher sind nicht für Kunden verfügbar. Sollten Sie weitere Fragen bezüglich Reparaturen haben, wenden Sie sich bitte an Ihre nächste Hewlett-Packard Serviceniederlassung.

Benutzerberatung

Sollten beim Einsatz Ihres Gerätes in bestimmten Anwendungsfällen Fragen auftauchen, so wenden Sie sich an den HP autorisierten Fachhändler, bei dem Sie das Gerät bezogen haben.

Bezugsquellennachweis über den Fachhandel erhalten Sie in der Bundesrepubblik Deutschland über:

Hewlett-Packard GmbH
Vertriebszentrale
Werbeabteilung Berner Str. 117
Postfach 560140

D–6000 Frankfurt 56

Produktinformation

Weitere Produktinformation finden Sie in der Zubehörbroschüre, die Ihrem Gerät beiliegt, bzw. bei Ihrem Hewlett-Packard Händler.

Anhang C

Tastenfeld-Operationen

Inhalt

Tastenfunktionen

Schreibmaschinentasten

[A] bis [Z] — Buchstaben.

[0] bis [9] — Ziffern.

[ ] — Leerzeichen.

[.] — Punkt. Wird als Dezimalpunkt in Zahlen und als letztes Zeichen in Abkürzungen verwendet.

[+] [-] [*] [/] [^] [\] ([CTL] [/]) — Arithmetische Symbole: Addition, Subtraktion, Multiplikation, Division, Potenzierung und Ganzzahldivision.

[,] — Komma. Zur Trennung von Parametern in Befehlen, Anweisungen und Funktionen benutzt.

[(] [)] — Klammern. Zur Eingabe von numerischen Ausdrücken benutzt.

[!] — Ausrufungszeichen. Für Kommentare am Zeilenende in Programmanweisungen und für Terminnotizen verwendet.

["] ['] — Doppelte und einfache Anführungszeichen. Zur Kennzeichnung von Filenamen und anderen Literalen.

[#] — Nummernzeichen. Zur Spezifikation von BASIC Filenummern in den Anweisungen `ASSIGN #`, `PRINT #`, `RESTORE #` und `READ #` und zur Zuweisung von Timernummern in den Anweisungen `ON TIMER #` und `OFF TIMER #` benutzt. Dient ebenfalls als Ungleichheitsoperator in Vergleichsabfragen.

[$] — Dollarzeichen. Dient zur Spezifikation von Stringvariablen und Stringfunktionen.

[&] — Dient zur Verkettung von Stringausdrücken.

[[] []] — Eckige Klammern. Dienen zur Dimensionierung von Stringvariablen und zur Spezifikation von Teilstrings.

@	Dient zur Bildung von Mehrfachanweisungszeilen.
;	Semikolon. Dient zur Trennung von Datenelementen in den Anweisungen PRINT, DISP, INPUT, PRINT # und READ #.
=	Gleichheitszeichen. Dient zur Wertzuweisung auf Variablen und für Vergleichsabfragen.
<	Kleinerzeichen. Wird in Vergleichsabfragen benutzt.
>	Größerzeichen. BASIC-Eingabeaufforderung für Programme und APPT-Befehle. Wird auch in Vergleichsabfragen benutzt.
:	Doppelpunkt. Text-Eingabeaufforderung. Dient auch zur Begrenzung von Einheitscodes.
%	Prozentzeichen.
?	Fragezeichen. Vorgegebene Eingabeaufforderung für die INPUT Anweisung.

Editiertasten

SHIFT	Umschalttaste. Gleichzeitig gedrückte Tasten führen ihre Alternativfunktion aus. Im Großschrift-Modus erzeugen gleichzeitig gedrückte Buchstabentasten Kleinbuchstaben.
CTL	Kontrolltaste. Erzeugt spezielle Anzeigezeichen und dient zum Aufbau einer Vielzahl von Tastenkombinationen.
LOCK	Dient zur Fixierung des Tastenfelds auf Großschrift und zur Aktivierung des numerischen Tastenfelds.
← →	Linkspfeil und Rechtspfeil. Bewegen den Cursor in der Anzeige und bewirken gegebenenfalls ein Verschieben der Information in der Anzeige.
↑ ↓	Aufpfeil und Abpfeil. Bewegen den Cursor in BASIC-, Text- und Terminfiles, und in den Verzeichnissen des Systems und der Massenspeicher aufwärts und abwärts.
I/R	Einfügungs/Ersetzungstaste. Tauscht den Ersetzungscursor und den Einsetzungscursor untereinander aus.
CLR	Löschtaste. Löscht die Anzeige.
BACK	Rückschritt-Taste. Setzt den Cursor um eine Stelle zurück, und löscht das dort stehende Zeichen.
DEL	Zeichenlöschtaste. Löscht ein Zeichen und verschiebt die nachfolgenden Zeichen in der Anzeige nach links.
TAB	Tabulator. Bewegt den Cursor über die Anzeigefelder in TIME- und APPT-Modus.

Systemtasten

Die Systemtasten und die Tasten ↑ und ↓ senden einen Wagenrücklauf/Zeilenvorschub an die Anzeige und an die DISPLAY IS Einheiten, bevor die eigentliche Funktion ausgeführt wird.

ATTN	Attention. Schaltet den HP-75 ein; unterbricht Programme, Listings, die automatische Zeilennumerierung, Kartenleseroperationen und HP-IL Operationen. Bestätigt fällige Termine.
TIME	Time. Schaltet den HP-75 in den TIME-Modus.
APPT	Appointment. Schaltet den HP-75 in den APPT-Modus. Nicht bestätigte Termine werden angezeigt.

[EDIT] Edit. Schaltet den HP-75 in den EDIT-Modus.

[FET] Fetch. Dient im EDIT-Modus als Eingabehilfe für den Befehl FETCH.

[RTN] Return. Der Ausdruck, die Anweisung oder der Befehl in der Anzeige wird ausgewertet, gespeichert oder ausgeführt.

[RUN] Run. Startet im EDIT-Modus das momentane BASIC-Programm. Bearbeitet im APPT-Modus fällige Termine.

Tastenkombinationen

System-Tastenkombinationen

[SHIFT] [CTL] [CLR] Bewirkt ein Zurücksetzen des Systems (siehe Abschnitt 1, Seite 13).*

[SHIFT] [ATTN] Bewirkt ein Ausschalten der Anzeige und die Desaktivierung aller Tasten außer [ATTN].

[SHIFT] [LOCK] Bewirkt ein Umschalten auf Großbuchstaben. Nochmaliges Drücken von [LOCK] bewirkt ein Zurückschalten auf Kleinschreibung.

[CTL] [LOCK] Aktiviert das numerische Tastenfeld. Nochmaliges Drücken von [LOCK] stellt das normale Tastenfeld wieder her.

[SHIFT] [FET] Zeigt die letzte ERROR, WARNING, Kartenleser-, Programmverfolgungs- oder HP-IL Meldung nochmals an, solange die Taste [FET] gedrückt bleibt. Die Meldung wird nach [CLR], [ATTN], [TIME], [APPT], [EDIT], [FET], [↑], [↓], [RTN] oder [RUN] gelöscht.

[SHIFT] [RUN] Führt ein Programm, bei der momentanen Zeile beginnend, in Einzelschritten aus.

Editier-Tastenkombinationen

[SHIFT] [↑] Stellt den Filepointer auf die erste Zeile des momentanen Files, auf den ersten Eintrag eines CAT ALL oder Massenspeicher-Verzeichnisses, oder auf den ersten Termin im APPT-Modus.

[SHIFT] [↓] Stellt den Filepointer auf die letzte Zeile des momentanen Files, auf den letzten Eintrag eines CAT ALL oder Massenspeicher-Verzeichnisses, oder auf den letzten Termin im APPT-Modus.

[CTL] [FET] Bringt im EDIT-Modus die letzte Eingabe zur Anzeige und macht sie editierbereit. Die Eingabe wird intakt angezeigt, wenn nach dem letzten [RTN] keine weiteren Tasten gedrückt worden sind. Bei den folgenden Befehlen wird der Eingabebuffer benötigt, und die letzte Eingabe ist somit verloren: FETCH, FETCH KEY, LIST, LOCK, PLIST und TRANSFORM.

[SHIFT] [DEL] Löscht im EDIT- und im TIME-Modus die Zeile von der momentanen Cursorposition bis zum Zeilenende. Löscht im APPT-Modus den derzeit angezeigten Termin, oder ersetzt ihn durch einen editierten Termin.

[SHIFT] [TAB] Verschiebt den Cursor nach links über die Anzeigefelder; Umkehr der Wirkung von [TAB].

[SHIFT] [←] Setzt den Cursor an den Anfang der Anzeigezeile.

---

* Wenn sie länger als eine Sekunde gedrückt werden, setzen [SHIFT] [CTL] [M], [I], [J], [8] und [I/R] das System zurück.

SHIFT → Setzt den Cursor ans Ende der Anzeigezeile.

CTL ← Verschiebt den Cursor um 32 Positionen nach links, jedoch nicht über den Anfang der Anzeigezeile hinaus.

CTL → Verschiebt den Cursor um 32 Positionen nach rechts, jedoch nicht über das Ende der Anzeigezeile hinaus.

Tastenkombination für Anzeigezeichen

SHIFT I/R Das der nächsten Taste oder Tastenkombination zugeordnete Zeichen wird angezeigt, unabhängig von der derzeitigen Tastendefinition.

Escape-Tastenkombination

CTL BACK Erzeugt den Dezimalcode 27 (ESC) als Anfangszeichen von Escapecodefolgen.

Tastenkombinationen für Anzeigeeinheiten

CTL ↑ Sendet ein ESC T (Rollen nach oben um eine Zeile) an die DISPLAY IS Einheiten (wird vom Anzeigefenster des HP-75 ignoriert).

CTL ↓ Sendet ein ESC S (Rollen nach unten um eine Zeile) an die DISPLAY IS Einheiten (wird vom Anzeigefenster des HP-75 ignoriert).

Anhang D

Referenztabellen

Inhalt

Der Zeichensatz

Die Funktion CHR$ gibt das Anzeigezeichen eines gegebenen Dezimalcodes von 0 bis 255 zurück. Die Argumente werden auf ganzzahlige Werte gerundet und in den richtigen Bereich transformiert (modulo 256). Der Dezimalcode eines unterstrichenen Anzeigezeichens ist um 128 größer als der des gleichen, nicht unterstrichenen Zeichens.

Die Funktion NUM erzeugt den Dezimalcode des ersten Zeichens eines gegebenen Zeichenstrings.

Beispiele: Drücken Sie nach jeder Eingabe RTN.

```
>chr$(65), chr$(65+128)
```

```
A                A
```

```
>num('a'), chr$(num('a')+128)
```

```
97               a
```

Zeichen mit den Dezimalcodes von 32 bis 126 sind Standard-Druckzeichen nach der ASCII (American Standard for Information Interchange) Konvention zugeordnet. Die Reaktion von Peripherieeinheiten auf die Dezimalcodes von 0 bis 31 und von 127 bis 255 finden Sie im Benutzerhandbuch der entsprechenden HP-IL Einheiten. Wenn Sie Sonderzeichen (zum Beispiel das Zeichen Formfeed, FF) darstellen wollen, sollten Sie nicht das Anzeigezeichen selbst benutzen, – Sie müssen sonst mit unerwartetem Verhalten der Peripherieeinheit bei Programmlistings rechnen; verwenden Sie stattdessen die Funktion CHR$ (zum Beispiel CHR$(12) an Stelle von μ).

Ein Stern (*) deutet an, daß Sie zuerst SHIFT I/R drücken müssen, um das mit der Taste oder Tastenfolge verknüpfte Zeichen anzuzeigen. Wenn keine Tastenfolge angegeben ist, kann das entsprechende Zeichen nur mit dem Befehl CHR$ erzeugt werden.

Dezimal-code	Anzeige-zeichen	Tastenfolge
0	▵	CTL Leertaste
1	▫	CTL A
2	⊠	CTL B
3	⇦	CTL C
4	α	CTL D

Dezimal-code	Anzeige-zeichen	Tastenfolge
128	▵	* ATTN
129	▫	* TIME
130	⊠	* APPT
131	⇦	* EDIT
132	α	* ↑

Dezimal-code	Anzeige-zeichen	Tastenfolge
5	⎕	CTL E
6	⌈	CTL F
7	⌊	CTL G
8	BS	CTL H
9	∘	CTL I
10	LF	CTL J
11	∧	CTL K
12	⍵	CTL L
13	CR	CTL M
14	⍫	CTL N
15	⍎	CTL O
16	⍟	CTL P
17	⍵	CTL Q
18	⍴	CTL R
19	∈	CTL S
20	⍨	CTL T
21	⍸	CTL U
22	⍷	CTL V
23	⍹	CTL W
24	⍤	CTL X
25	⍺	CTL Y
26	⍶	CTL Z
27	ESC	CTL BACK
28	⍳	CTL +
29	≠	CTL =
30	⍣	CTL ;
31	▒	CTL 8
32		space bar
33	!	SHIFT 1
34	"	SHIFT 2
35	#	SHIFT 3
36	$	SHIFT 4
37	%	SHIFT 5
38	&	SHIFT 6
39	'	SHIFT 7
40	(	SHIFT 8
41	)	SHIFT 9
42	*	*
43	+	+
44	,	,
45	-	-
46	.	.

Dezimal-code	Anzeige-zeichen	Tastenfolge
133	⎕	* ↓
134	⌈	* ←
135	⌊	* →
136	∘	* I/R
137	∧	* FET
138	⍵	* DEL
139	⍫	* CLR
140	⍎	* LOCK
141	▒	* RUN
142	⍫	* TAB
143	⍎	* BACK
144	⍟	
145	⍵	
146	⍴	
147	∈	
148	⍨	
149	⍸	
150	⍷	
151	⍹	
152	⍤	
153	⍺	
154	⍶	
155	⍳	
156	≠	
157	⍣	
158	⍱	
159	▒	
160		* SHIFT ATTN
161	¡	* SHIFT TIME
162	"	* SHIFT APPT
163	#	* SHIFT EDIT
164	$	* SHIFT ↑
165	%	* SHIFT ↓
166	&	* SHIFT ←
167	'	* SHIFT →
168	(	* SHIFT I/R
169	)	* SHIFT FET
170	*	* SHIFT DEL
171	+	* SHIFT CLR
172	¿	* SHIFT LOCK
173	=	* SHIFT RUN
174	¡	* SHIFT TAB

Dezimal-code	Anzeige-zeichen	Tastenfolge
47	/	/
48	0	0
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	SHIFT ;
59	;	;
60	<	SHIFT ,
61	=	=
62	>	SHIFT .
63	?	SHIFT /
64	@	SHIFT +
65	A	SHIFT A
66	B	SHIFT B
67	C	SHIFT C
68	D	SHIFT D
69	E	SHIFT E
70	F	SHIFT F
71	G	SHIFT G
72	H	SHIFT H
73	I	SHIFT I
74	J	SHIFT J
75	K	SHIFT K
76	L	SHIFT L
77	M	SHIFT M
78	N	SHIFT N
79	O	SHIFT O
80	P	SHIFT P
81	Q	SHIFT Q
82	R	SHIFT R
83	S	SHIFT S
84	T	SHIFT T
85	U	SHIFT U
86	V	SHIFT V
87	W	SHIFT W
88	X	SHIFT X

Dezimal-code	Anzeige-zeichen	Tastenfolge
175	↙	
176	0	* CTL 0
177	1	* CTL 1
178	2	* CTL 2
179	3	* CTL 3
180	4	* CTL 4
181	5	* CTL 5
182	6	* CTL 6
183	7	* CTL 7
184	8	* CTL 8
185	9	* CTL 9
186	↕	
187	↓	
188	≤	
189	≡	
190	≥	
191	?	
192	@	* CTL ATTN
193	A	* CTL TIME
194	B	* CTL APPT
195	C	* CTL EDIT
196	D	* CTL ↑
197	E	* CTL ↓
198	F	* CTL ←
199	G	* CTL →
200	H	* CTL I/R
201	I	* CTL FET
202	J	* CTL DEL
203	K	* CTL CLR
204	L	* CTL LOCK
205	M	* CTL RUN
206	N	* CTL TAB
207	O	
208	P	
209	Q	
210	R	
211	S	
212	T	
213	U	
214	V	
215	W	
216	X	

Dezimal-code	Anzeige-zeichen	Tastenfolge
89	Y	SHIFT Y
90	Z	SHIFT Z
91	[	SHIFT 0
92	\	CTL /
93	]	SHIFT .
94	^	SHIFT -
95	_	CTL -
96	`	CTL 7
97	a	A
98	b	B
99	c	C
100	d	D
101	e	E
102	f	F
103	g	G
104	h	H
105	i	I
106	j	J
107	k	K
108	l	L
109	m	M
110	n	N
111	o	O
112	p	P
113	q	Q
114	r	R
115	s	S
116	t	T
117	u	U
118	v	V
119	w	W
120	x	X
121	y	Y
122	z	Z
123	{	CTL ,
124	|	SHIFT =
125	}	CTL ·
126	~	CTL *
127	⊢	CTL 9

Dezimal-code	Anzeige-zeichen	Tastenfolge
217	Y	
218	Z	
219	[	
220	\	
221	]	
222	^	
223	_	
224	`	* SHIFT CTL ATTN
225	a	* SHIFT CTL TIME
226	b	* SHIFT CTL APPT
227	c	* SHIFT CTL EDIT
228	d	* SHIFT CTL ↑
229	e	* SHIFT CTL ↓
230	f	* SHIFT CTL ←
231	g	* SHIFT CTL →
232	h	* SHIFT CTL I/R
233	i	* SHIFT CTL FET
234	j	* SHIFT CTL DEL
235	k	* SHIFT CTL CLR
236	l	* SHIFT CTL LOCK
237	m	* SHIFT CTL RUN
238	n	* SHIFT CTL TAB
239	o	* SHIFT CTL BACK
240	p	
241	q	
242	r	
243	s	
244	t	
245	u	
246	v	
247	w	
248	x	
249	y	
250	z	
251	{	
252	|	
253	}	
254	~	BYE†
255	⊢	

† Wenn einer Taste der Dezimalcode 254 zugewiesen wird, erzeugt diese Taste den Befehl BYE. Wenn das Anzeigezeichen 254 mit CHR$ erzeugt wird, erhalten Sie das Zeichen ~ in der Anzeige.

Systemspeicheranforderungen

Sie können die Größe eines initialisierten Programmes bestimmen, indem Sie zuerst die Ausführung unterbrechen (mit [ATTN]) und den verfügbaren Speicher bestimmen (mit `mem` [RTN]). Deallokatisieren Sie danach das Programm, indem Sie eine Zeile editieren. Der dadurch verfügbar gemachte zusätzliche Speicherplatz plus die Größe des deallokatisierten Files (wie im Katalogeintrag angegeben) ergeben die Gesamtgröße.

Speicherobjekt	Benötigter Speicherplatz
Termine	15 Bytes für den Terminfile +7 Bytes/Termin +1 Byte/Zeichen im Notizfeld `NOTE` +5 Bytes/Wiederholungstermin.
Datenelemente	5 Bytes/`DATA` Anweisung +4 Bytes/`INTEGER` Wert +9 Bytes/`REAL`-Wert +2 Bytes/Datenelement +1 Byte/Zeichen im String.
Datenpointer	15 Bytes/Datenpointer, der mit `ASSIGN #` erzeugt worden ist.
Files (BASIC und Text)	15 Bytes/File +3 Bytes/Zeile +1–3 Bytes/Schlüsselwort (BASIC) oder +1 Byte/Zeichen (Text).
Files (LEX und Austausch-)	Siehe Katalogeintrag des Files
HP-IL Zuweisungen	7 Bytes/deklarierte Einheit (in `ASSIGN IO`).
Tastendefinitionen	3 Bytes/Definition +1 Byte/Zeichen im Tastendefinitionsstring +1 Byte für das End-Semikolon.
Massenspeicherbefehle `CAT`, `COPY`, `PURGE`, `RENAME`	43 Bytes/Befehl.
Massenspeichereinheiten	105 Bytes/Einheit.
`PACK` Befehl	256 Bytes +6 Bytes/File.
Einsteck-ROMs	Siehe Benutzerhandbuch des ROMs.
Programmaufrufe	30 Bytes/`CALL` Anweisung +2 Bytes/Variable des aufrufenden Programms +Anzahl Bytes, die die Variablengenauigkeit erfordert.
Timer	63 Bytes/Timer (der in einer `ON TIMER` Anweisung gesetzt wurde) +ein oder mehr Bytes pro Timeranweisung.
`TRANSFORM` Befehl	Maximal 255 Bytes während der Umformung.
Variablen (Rechner- und initialisierte Programmvariablen)	
Einfache numerische Variablen	
`REAL`	12 Bytes/Variable.
`SHORT`	8 Bytes/Variable.
`INTEGER`	7 Bytes/Variable.
Numerische Feldvariablen	10 Bytes/Feldvariable plus
`REAL`	8 Bytes/Element.
`SHORT`	4 Bytes/Element.
`INTEGER`	3 Bytes/Element.
Stringvariablen	8 Bytes/Variable + dimensionierte Länge.

Globale und lokale Deklarationen

Eine *globale* Deklaration ist eine Einstellung des HP-75, die erhalten bleibt, bis:

- die Einstellung mit einer neuen Deklaration über das Tastenfeld oder in einem Programm verändert wird,
- der HP-75 zurückgesetzt wird.

Eine *lokale* Deklaration beeinflußt nur die Tastenfeldberechnungen oder das Programm, in dem sie auftritt.

Lokal	Global
DATA	ALARM ON, ALARM OFF
DEF FN, END DEF	ASSIGN IO, OFF IO, RESTORE IO
DIM	ASSIGN #
IMAGE	BEEP OFF, BEEP ON
LET	DEF KEY
ON ERROR, OFF ERROR	DEFAULT OFF, DEFAULT ON
ON TIMER #	DELAY
OPTION BASE	DISPLAY IS, PRINTER IS
REAL, SHORT, INTEGER	ENDLINE
	LOCK
	MARGIN
	OFF TIMER #
	OPTION ANGLE RADIANS
	OPTION ANGLE DEGREES
	STANDBY OFF, STANDBY ON
	TRACE FLOW, TRACE VARS, TRACE OFF
	WIDTH, PWIDTH

Anzeige-Escapecodes

ESC symbolisiert das Escapezeichen, Dezimalcode 27. Wenn Sie ESC durch CTL BACK oder CHR$(27) anzeigen, wird ein Escapezeichen an die Anzeige des HP-75 und an die anderen DISPLAY IS Einheiten gesandt. Wenn CHR$(27) ausgedruckt wird, wird ein Escapezeichen an die PRINTER IS Einheiten gesandt. Die Anzeige des HP-75 und die meisten HP-IL Einheiten zeigen spezielle Reaktionen auf vordefinierte *Escapecodes*; das sind Anzeige- oder Druckeranweisungen, die aus dem Escapezeichen und den direkt darauf folgenden Zeichen bestehen. Ein gegebener Escapecode kann von einer Einheit ignoriert werden, während er von einer anderen als Systembefehl verstanden wird. Ein ESC C zum Beispiel wird vom Thermodrucker HP82162A ignoriert, das Video-Interface HP82163 bewegt dagegen den Cursor um eine Stelle nach rechts. Ein ESC C kann zum Beispiel mit der Eingabe DISP CHR$(27)&'C' an die Anzeige des HP-75 und an andere Anzeigeeinheiten gesandt werden.

Die Anzeige des HP-75 reagiert auf 12 Escapecodes. Alle anderen Escapecodes, die in den DISP oder PRINT Anweisungen benutzt werden können, werden von der Anzeige ignoriert.

Die 32 Fensterpositionen werden durch die Nummern 0 bis 31 identifiziert.

Escapecode	Instruktion	Beschreibung
ESC C	Cursor nach rechts*	Bewegt den Cursor um eine Stelle nach rechts.
ESC D	Cursor nach links*	Bewegt den Cursor um eine Stelle nach links.
ESC E	Lösche alles*	Setzt den Cursor auf Pos. 0 und löscht die Anzeige.
ESC G	Cursor Return	Setzt den Cursor auf Pos. 0.
ESC H	Cursor Home*	Setzt den Cursor auf Pos. 0.
ESC J	Lösche nach Cursor*	Löscht die Anzeige ab der momentanen Cursorposition.
ESC K	Lösche bis Zeilenende	Löscht die Anzeige ab der momentanen Cursorposition.
ESC O	Lösche Zeichen mit Rundumverschiebung	Löscht das Zeichen an der Position des Cursors und schiebt alle nachlaufenden Zeichen mit Rundumverschiebung nach links.
ESC P	Lösche Zeichen	Löscht das Zeichen an der Position des Cursors und schiebt alle nachlaufenden Zeichen nach links.
ESC <	Cursor aus*	Schaltet den Cursor aus.
ESC >	Cursor ein*	Schaltet den Cursor ein.
ESC %*cr*	Cursor an Adresse*	Bewegt den Cursor an die vom ersten Anzeigezeichen modulo 32 spezifizierte Anzeigeposition. Der Dezimalcode des zweiten Zeichens wird vom Video-Interface als Zeilenparameter benutzt, vom HP-75 aber ignoriert. Mit DISP CHR$(27);'%';CHR$(16);CHR$(8);'hier' zum Beispiel wird der Cursor auf Spalte 16 (und Zeile 8 des Video-Interface) gesetzt und von dieser Position an hier angezeigt.

Ein Stern (*) deutet an, daß das Video-Interface HP82163 vergleichbar reagiert, wenn es als DISPLAY IS Einheit erklärt worden ist. Schlagen Sie auch im Benutzerhandbuch des Video-Interface nach.

Geben Sie nach einem Escapecode ein Semikolon ein, um den Wagenrücklauf/Zeilenvorschub zu unterdrücken, der normalerweise eine DISP oder PRINT Anweisung beendet.

Mit SHIFT ↑ erzeugen Sie ein ESC T (Rollen nach oben um eine Zeile) und mit SHIFT ↓ ein ESC S (Rollen nach unten um eine Zeile) auf DISPLAY IS Einheiten. Das Anzeigefenster des HP-75 ignoriert diese Codes.

System-Voreinstellungen

Bedingung	Nach einem Zurücksetzen	Beim Einschalten nach einem automatischen Ausschalten
ALARM	ON	Vorherige Einstellung.
ASSIGN IO	Keine HP-IL Zuweisungen.	Vorherige HP-IL Zuweisungen.
ASSIGN #	Keine Zuweisung von Filenummern.	Vorherige Zuweisung der Filenummern.
BEEP	ON	Vorherige Einstellung.
CTL FET	Zeitsetzmaske.	Leere Zeile.
DEFAULT	ON	Vorherige Einstellung.
DEF KEY	Identitätszuweisung.	Vorherige Tastendefinitionen.
DELAY	1 Sekunde.	Vorherige Einstellung.
DISPLAY IS	* (Anzeige des HP-75).	Vorherige Anzeigeeinheiten.
ENDLINE	CR/LF.	Vorherige Einstellung.
ERRL	0	Vorherige Fehlerzeile.
ERRN	0	Vorherige Fehlernummer.
EXACT	EXACT Marken sind gesetzt.	EXACT Marken sind unverändert.
FETCH	Zeile 0 des Arbeitsfiles.	Vorherige anstehende Zeile.
Files	BASIC Arbeitsfile (0 Bytes, Zeit und Datum der Uhr.)	Vorherige BASIC Files, Textfiles, APPT-Files, Keys-Files, LEX Files, LIF1 Files.
Tastenfeld	Nicht umgeschaltet.	Vorheriger Tastenfeldmodus.
LOCK	Kein Codewort.	Vorheriges Codewort.
MARGIN	91	Vorherige Einstellung.
Modus	TIME	EDIT
PRINTER IS	* (Anzeige des HP-75).	Vorherige Druckereinheiten.
PWIDTH	32 Spalten.	Vorherige Einstellung.
RES	0	Letztes numerisches Ergebnis.
RND	.529199358633	Nächste Zahl in der Folge.
SHIFT FET	Leere Anzeige.	Leere Anzeige.
STANDBY	OFF	Vorherige Einstellung.
STATS Maske	MDY, ~, AM, YEAR	Vorherige Einstellungen.
TIME Anzeige	Zeitsetzwerte.	Aktuelle Zeit.
Trigonometrischer Modus	OPTION ANGLE RADIANS	Vorherige Einstellung.
Variablen:		
Rechner-	Keine.	Vorherige Zuweisungen.
Programm-	Keine.	Vorherige Zuweisungen bei initialisiertem Programm.
WIDTH	32 Spalten	Vorherige Einstellung.

Abkürzungen

Die folgenden Abkürzungen stellen die kürzesten vom HP-75 noch getrennt erkennbaren Kurzformen dar. Längere Abkürzungen sind erlaubt, solange der Punkt nicht für das letzte Zeichen des Schlüsselwortes steht. Abkürzungen dürfen im Wort selbst keine Leerzeichen enthalten.

Bei Listings und beim Abrufen von Zeilen schreibt der HP-75 die Schlüsselworte ganz aus.

Schlüsselwort	Abkürzung
ACOS()	ac.()
ALARM OFF	al.off
ALARM ON	al.on
ANGLE()	ang.()
ASSIGN #	as.#
ASSIGN IO	as..
AUTO	a.
BEEP	be.
BEEP OFF	be.off
BEEP ON	be.on
BYE	b.
CAT$()	c.()
CAT ALL	c..
CEIL()	ce.()
CHR$()	ch.()
CLEAR *Einheitscode*	cl.':*ec*'
CLEAR LOOP	cl.l.
CLEAR VARS	cl..
COPY	co.
DATA	da.
DEFAULT OFF	defa.off
DEFAULT ON	defa.on
DEF KEY	d..
DELAY	d.
DELETE	dele.
DISP	di.
DISPLAY IS	di..
DISP USING	di.us.
EDIT	e.
EDIT BASIC	e.ba.
EDIT TEXT	e.t.
ELSE	el.
END DEF	end d.
ENDLINE	en.
ERRN	er.
FETCH	f.
FETCH KEY	f..
FLOOR()	fl.()
GOSUB	gos.
GOTO	g.

Schlüsselwort	Abkürzung
IMAGE	im.
INITIALIZE	ini.
INPUT	i.
INTEGER	int.
KEY$	k.
LIST	l.
LIST IO	l..
LET FN	le..
LOG10()	lo.()
MARGIN	ma.
MERGE	m.
NAME	n.
NEXT	ne.
OFF ERROR	of.e.
OFF TIMER #	of.t.#
ON ERROR	o..
ON TIMER #	o.t.#
OPTION ANGLE DEGREES	op.a.d.
OPTION ANGLE RADIANS	op.a.r.
OPTION BASE	op..
PACK	pa.
PLIST	pl.
POP	p.
PRINT	pri.
PRINT #	pri.#
PRINTER IS	pri..
PRINT USING	pri.us.
PROTECT	pr.
PURGE	pu.
PWIDTH	pw.
RANDOMIZE	ra.
READ #	r.#
REAL	re.
RENAME	ren.
RENUMBER	renu.
RESTORE	res.
RESTORE #	res.#
RESTORE IO	res..
RETURN	ret.
RUN	r

Schlüsselwort	Abkürzung
SHORT	sh.
STANDBY OFF	s.off
STANDBY ON	s.on
THEN	th.
TIME$	ti.
TRACE FLOW	t.
TRACE OFF	tr.o.
TRACE VARS	tr.v.
TRANSFORM	tr.

Schlüsselwort	Abkürzung
UNPROTECT	u.
UPRC$()	up.()
VER$	v.
WAIT	wa.
WIDTH	w.

Anhang E

Fehlerbedingungen

Inhalt

Auftreten eines Fehlers

Beim Auftreten eines Fehlers erzeugt der HP-75 ein Tonsignal, setzt die Statusanzeige **ERROR** und zeigt eine ERROR oder WARNING Meldung mit der derzeitigen Verzögerungsrate an. Wenn der Fehler während einer Programmausführung auftritt, wird auch die Nummer der den Fehler erzeugenden Zeile angezeigt. Mit [SHIFT] [FET] wird die Meldung noch einmal angezeigt, solange die Taste [FET] gedrückt gehalten wird. Mit errn [RTN] erhalten Sie die Identifikationsnummer des Fehlers oder der Warnung. Wenn Sie [CLR], [ATTN], [TIME], [APPT], [EDIT], [FET], [↑], [↓], [RTN] oder [RUN] drücken, wird die Statusanzeige zurückgesetzt und die Fehlermeldung gelöscht.

Nach einer WARNING gibt der HP-75 Ersatzwerte vor, und die Ausführung des Programms wird fortgesetzt (falls nicht ON ERROR deklariert worden ist). Eine Fehlerbedingung unterbricht die Programmausführung bei der den Fehler erzeugenden Anweisung. Wenn das unterbrochene Programm gleichzeitig der momentane File ist, wird der Filepointer auf die Zeile gestellt, in der der Fehler aufgetreten ist; mit [FET] und dann [RTN] können Sie die Zeile zur Anzeige bringen. Das Programm bleibt initialisiert, bis Sie eine Zeile editieren oder ein anderes Programm starten.

Mathematische Warnungen und Fehler		
Nummer	**Meldung und Ursache**	**Ersatzwert***
1	`num too small` Zahl liegt zwischen ±`EPS`.	`0`
2	`num too large` • größer als ±`INF`. • größer als in `SHORT` möglich. • größer als in `INTEGER` möglich.	±9.99999999999E499 ±9.9999E99 ±99999
3	`COT or CSC inf` `COT` oder `CSC` ist $n \times 180°$; n = ganzzahlig.	9.99999999999E499
4	`TAN or SEC inf` `TAN` oder `SEC` ist $n \times 90°$, n ganzzahlig. und ungerade.	9.99999999999E499
5	`0^neg` Null zu einer negativen Potenz.	9.99999999999E499
6	`0^0` Null hoch null.	`1`
7	`no value.` • Der Wert einer Stringvariablen wird abgerufen, bevor ein Wert zugewiesen wurde. • Der Wert einer einfachen numerischen Variablen oder eines numerischen Feldelements wird abgerufen, bevor ein Wert zugewiesen wurde.	`" "` `0`
8	`/zero` Division durch Null.	±9.99999999999E499
Die übrigen mathematischen Fehler besitzen keine Ersatzwerte.		
9	`neg^non-integer` Negativwert zu einer negativen Potenz.	
10	`SQR(neg number)` Quadratwurzel einer negativen Zahl.	
11	`arg out of range` Argument zu groß oder zu klein.	
12	`LOG(0)` Logarithmus von Null.	
13	`LOG(neg number)` Logarithmus einer negativen Zahl.	

* bei `DEFAULT ON`.

Systemfehler (14-18)	
Nummer	**Meldung und Ursache**
14	`Low batteries` Ersetzen Sie die Batterien oder stecken Sie das Netzadapter/Ladegerät ein.
15	`system error` Der HP-75 muß evtl. zurückgesetzt werden.
16	`not enough memory` • nicht genügend Speicherplatz verfügbar, um einen Magnetkarten- oder Bandkassettenfile in den Speicher einzulesen. • Ein File oder Programm benötigt zu viel Speicher. Mögliche Lösungen: • Löschen Sie eine oder mehrere Zeilen des momentanen Files. • Führen Sie `CLEAR VARS` aus, um die Rechnervariablen zu löschen. • Verkleinern Sie die Genauigkeit oder Dimensionen von Variablen. • Löschen Sie einen oder mehrere Files aus dem Speicher. • Spezifizieren Sie unbenutzte Datenfilepointer in `ASSIGN # TO *`. • Führen Sie `CATALL` aus und drücken Sie dann ↑ oder ↓, um nicht mehr benötigte Files zu finden, und löschen Sie diese mit EDIT `PURGE`. • Desaktivieren Sie die HP-IL Schleife mit `ASSIGN IO *`.
17	`RAM is invalid` • Zeigt einen defekten Schaltkreis an. Das Gerät muß gegebenenfalls repariert werden. • Zeigt einen Speicherverlust an.
18	`ROM missing` Ein notwendiges Einsteck-ROM oder LEX-File fehlt.

Warnungen des Kartenlesers (19-26)	
Nummer	**Meldung und Ursache**
19	`write protected` Versuch, eine schreibgeschützte Magnetkarte zu beschreiben. Führen Sie zuerst `UNPROTECT` durch.
20	`not this file` Eine Spur gehört nicht zum derzeit eingelesenen File.
21	`verify failed` Eine Spur muß noch zwei weitere Male durch den Magnetkartenleser geführt werden, einmal zum Beschreiben der Spur und einmal zur Prüfung. Reinigen Sie die Magnetkarte zuerst.
22	`unknown card` Die Information auf der Spur wird vom HP-75 nicht erkannt.
23	`bad read/write` Ein Fehlversuch, eine Spur richtig zu beschreiben oder einzulesen. Reinigen Sie die Magnetkarte und versuchen Sie es noch einmal.
24	`pulled too fast` Ziehen Sie die Magnetkarte langsamer durch den Magnetkartenleser.
25	`pulled too slow` Ziehen Sie die Magnetkarte schneller durch den Magnetkartenleser.
26	`wrong name` Der Filespezifikator in einem `COPY` Befehl stimmt mit dem Namen des Magnetkartenfiles nicht überein.

Programmfehler (27-54)	
Nummer	**Meldung und Ursache**
27	`invalid subscript` • Ein Feldindex liegt außerhalb des erlaubten Bereichs. • Ein Teilstring einer Stringvariabeln liegt außerhalb der dimensionierten Stringlänge.
28	`record overflow` Eine direkte `PRINT #` Anweisung bewegt den Datenpointer über das Zeilenende hinaus.
29	`ON ERROR overflow` Eine `ON ERROR` Routine sollte den Fehler 49 – `GOSUB overflow` – bearbeiten. Der HP-75 stoppt die Programmausführung und zeigt diese Meldung an, anstatt mit einem weiteren Unterprogramm den Fehler noch zu vergrößern.
30	`OPTION BASE` • Ein unzulässiger `OPTION BASE` Parameter wurde spezifiziert. • Die Anweisung `OPTION BASE` erscheint nach einem Bezug auf eine Feldvariable. • Mehrfache `OPTION BASE` Anweisungen in einem Programm.
31	`CONT before RUN` Es wurde vor der Initialisierung oder nach der Deallokation eines Programms `CONT` eingegeben oder SHIFT RUN gedrückt.
32	`missing line` • Eine Zeile, die mit `GOTO` oder `GOSUB` adressiert wird, ist nicht vorhanden. Eine Zeile, die in `DISP USING` oder `PRINT USING` adressiert wird, fehlt.
33	`data type` Eine `READ` oder `READ #` Anweisung versucht, Stringinformation in eine numerische Variable einzulesen.
34	`no data` • In einer `READ` oder `READ #` Anweisung (seriell) wird versucht, über das Zeilenende hinaus zu lesen. • Es wird eine `RESTORE` oder `RESTORE #` Anweisung ohne Daten im File ausgeführt. • Eine existierende Zeile, die nicht das Schlüsselwort `DATA` enthält, wird in einer der Anweisungen `RESTORE`, `RESTORE #`, `READ #` oder `PRINT #` (direkter Zugriff) spezifiziert.
35	`DIM exist var` Genauigkeitsdeklaration für eine Variable, die schon zuvor im Programm benutzt wurde.
36	`Invalid DIM` Unzulässiges Dimensionieren eines Feldes nach einer `OPTION BASE` Deklaration.
37	`duplicate FN` Der gleiche Funktionsname erscheint in zwei oder mehreren `DEF FN` Anweisungen.
38	`no END DEF` Eine benutzerdefinierte Funktion mit mehreren Zeilen enthält keine `END DEF` Anweisung.
39	`FN missing` Versuchte Verzweigung in die Mitte einer mehrzeiligen benutzerdefinierten Funktion.

Programmfehler (27-54) (*Fortsetzung*)	
Nummer	**Meldung und Ursache**
40	`FN parameter` • die formale Parameterliste in der Funktionsdefinition stimmt nicht mit der gegebenen Parameterliste im Hauptprogramm überein. • Versuch, eine nicht vorhandene benutzerdefinierte Funktion aufzurufen.
41	`FN calls itself` Eine benutzerdefinierte Funktion ist rekursiv definiert.
42	`string too long` • Versuch, einer Stringvariablen zu viele Zeichen zuzuweisen. • Schreiben eines Strings mit mehr als 251 Zeichen in einen BASIC-File, oder mit mehr als 255 Zeichen in einen Textfile. Erzeugt eine Warnung und schneidet den String ab.
43*	`numeric input` • Versuch, bei einer `INPUT` Anweisung Alpha-Zeichen einzugeben. • Drücken von RTN, bevor alle erforderlichen Werte für die numerischen Variablen der Eingabeliste eingegeben worden sind.
44*	`too many inputs` Eingabe zu vieler Daten nach einer `INPUT` Anweisung.
45	`missing ASSIGN #` • Ausführung einer der Befehle `PRINT #, READ #, RESTORE #` ohne vorherige Zuweisung einer Filenummer.
46	`missing NEXT` Fehlendes `NEXT` in einer `FOR...NEXT` Schleife.
47	`no matching FOR` Ein `NEXT` ohne dazugehöriges `FOR` wurde angetroffen. Vermutlich unrichtig verschachtelte Schleifen.
48	`FOR overflow` Mehr als 255 Verschachtelungen von `FOR...NEXT` Schleifen.
49	`GOSUB overflow` Mehr als 255 Verschachtelungen von Unterprogrammen.
50	`RETURN w/o GOSUB` Ein `RETURN` ohne anstehende Rücksprungbedingung wurde angetroffen.
51	`PRINT# to runfile` Ein Programm bezieht sich in einer `PRINT #` Anweisung auf sich selbst.
52	`invalid IMAGE` Mindestens einer der Spezifikatoren in `IMAGE, DISP USING` oder `PRINT USING` ist unrichtig. Kann durch ein unzulässiges Zeichen im `IMAGE` String verursacht sein.
53	`invalid USING` Einer der Spezifikatoren in `IMAGE, DISP USING` oder `PRINT USING` kann ein anzuzeigendes oder zu druckendes Datenelement nicht darstellen.
54	`invalid TAB` Ein `TAB` Argument wird auf einen Wert kleiner 1 gerundet. Sie erhalten eine Warnung, und der Ersatzwert 1 wird zur Verfügung gestellt.

* Bei einem Eingabefehler fordert der HP-75 noch einmal zur Eingabe auf. Wenn bei einem `INPUT` Fehler `ON ERROR` deklariert ist, werden den `INPUT` Variablen so viele Werte wie möglich zugewiesen; danach wird die `ON ERROR` Anweisung ausgeführt.

HP-IL Fehler (55 bis 61)	
Nummer	**Meldung und Ursache**
55	`ASSIGN IO needed` Ausführung von `DISPLAY IS`, `PRINTER IS`, `LIST IO`, `OFF IO` oder `RESTORE IO` ohne vorhergehende Zuweisung der Einheiten in der Schleife.
56	`no loop response` Die HP-IL Schleife ist angeschlossen, aber keine Einheit reagierte auf den Befehl `ASSIGN IO`.
57	`bad transmission` • Hardware-Fehler. • Drücken von [ATTN] während einer Übertragung.
58	`loop timeout` • Eine Unterbrechung oder Stromausfall eines Peripheriegerätes während einer Schleifenübertragung. • Ein Gerät benötigt bei `STANDBY OFF` Einstellung mehr als 10 Sekunden, um eine einzelne HP-IL Instruktion auszuführen. Versichern Sie sich, daß alle Einheiten richtig angeschlossen und mit Strom versorgt sind, und führen Sie dann `RESTORE IO` aus. Falls die Unterbrechung beabsichtigt war, verbinden Sie die Buchsen IN und OUT des HP-75 mit einem HP-IL Kabel und führen Sie `OFF IO` aus. Setzen Sie `STANDBY ON`, falls eine Einheit länger als 10 Sekunden zur Reaktion auf einen Befehl benötigt.
59	`too many names` • Mehr Namen als Einheiten in `ASSIGN IO`. Erzeugt ein Warnung; es werden alle existierenden Geräte zugewiesen. • Nach einem `RESTORE` Befehl befinden sich mehr Einheitsnamen im Speicher als Einheiten in der Schleife gefunden wurden. Erzeugt einen Fehler; es werden keine Einheiten zugewiesen. Verbinden Sie die ursprünglichen Einheiten und führen Sie nochmals `RESTORE IO` aus, oder deklarieren Sie erneut mit `ASSIGN IO`.
60	`RESTORE IO needed` Der HP-75 versucht, nach einem `OFF IO` Befehl eine HP-IL Instruktion auszuführen.
61	`>31 devices` Es befinden sich einschließlich des HP-75 mehr als 31 Einheiten in der HP-IL Schleife.

File- und Einheitsfehler (62-69)	
Nummer	**Meldung und Ursache**
62	`file not found` Der spezifizierte File existiert nicht im Speicher oder auf dem Massenspeichermedium.
63	`invalid filespec` • Ungültiger Name für einen File im Speicher. Namen sind hier auf ein bis acht Zeichen beschränkt; davon muß das erste ein Buchstabe oder Punkt und die folgenden Buchstaben oder Ziffern sein. • Ungültiger Name für einen File auf Massenspeicher oder Magnetkarte. Namen sind hier auf ein bis maximal acht Zeichen beschränkt, das erste muß ein Buchstabe sein (Punkte sind nicht erlaubt). • Ungültiger HP-IL Einheitscode. Auf einen oder zwei Buchstaben oder einen Buchstaben und eine Ziffer beschränkt. • `TRANSFORM INTO LIF1` wird auf einen Arbeitsfile angewandt. • Ein Anzeige-Einheitscode wird in einem Massenspeicherbefehl spezifiziert.
64	`duplicate name` • Doppelter Filename. Es existiert schon ein File dieses Namens im Speicher oder auf einem Massenspeichermedium. • Doppelter Einheitscode in einer `ASSIGN IO` Deklaration. Wenn `ASSIGN IO` Einheiten nacheinander vom Tastenfeld aus zugewiesen werden, erhalten Sie eine Warnung; der HP-75 fordert einen anderen Einheitscode an.

File- und Einheitsfehler (62 bis 69) (*Fortsetzung*)	
Nummer	**Meldung und Ursache**
65	`access restricted` • Versuch, einen Termin-, Text-, LEX oder LIF1 File zu starten. • Versuch, einen privaten BASIC-File zu editieren, zu listen oder zu kopieren. • Versuchtes `PRINT #` oder `READ #` auf einen privaten BASIC-File, einen LEX File oder einen LIF1 File. • Versuch, mit `PRINT #` einen numerischen Wert in einen Textfile zu schreiben.
66	`invalid password` • Das Paßwort für einen Massenspeicher- oder Magnetkartenfile ist beim Kopieren des Files nicht korrekt spezifiziert. • Ein Paßwort wird beim Kopieren eines LIF1 Files oder eines nicht vom HP-75 erzeugten Files auf oder von einem Massenspeichermedium spezifiziert. Erzeugt eine Warnung; Kopieren ist möglich, aber es wird kein Paßwort angefügt.
67	`line too long` • Eine von einem File gelesene oder ausgelistete Zeile enthält mehr als drei Anzeigefenster lange Information. Erzeugt eine Warnung, und die ersten 94 Zeichen der Zeile werden angezeigt. Beim Drücken von [RTN] wird die abgerufene Zeile beim 94. Zeichen abgeschnitten, falls dies möglich ist; andernfalls wird ein Syntaxfehler gemeldet . • Abrufen einer Tastendefinition mit mehr als 80 Zeichen. Nur die ersten 80 Zeichen der Tastendefinition werden angezeigt. • Versuch, einen LIF1 File mit außergewöhnlich langen Zeilen in Text oder BASIC umzuwandeln; erzeugt diese Warnung und setzt `! ?` direkt hinter der Zeilennummer ein.
68	`wrong file type` • Versuch, einen Nichttextfile in `keys` umzubenennen. • Versuch, einen Nichtterminfile in `appt` umzubenennen. • Versuch, einen File in einen File eines anderen Typs zu mischen. • Versuch, einen Massenspeicherfile unbekannten Typs (`?`) in den Speicher einzulesen. • Versuch, einen Text-, Termin-, LEX oder LIF1 File auf einen Privatfile zu kopieren; erzeugt eine Warnung und eine nichtprivate Kopie.
69	`workfile name` Versuch, während der Editierung eines nichtleeren, unbenannten Arbeitsfiles einen anderen File zu editieren. Führen Sie die Operationen `NAME`, `RENAME` oder `PURGE` auf dem Arbeitsfile durch, bevor Sie den `EDIT` Befehl ausführen oder bei `CATALL` die Taste [EDIT] drücken.

Warnung im TIME-Modus (70)	
Nummer	**Meldung und Ursache**
70	`time adjust bad` Versuch, die Ganggeschwindigkeit um mehr als ±10% zu verändern. Erzeugt eine Warnung; der Anpassungsfaktor bleibt erhalten, und eine neue Anpassungsperiode beginnt.

Fehler im APPT-Modus (71 bis 77)	
Nummer	**Meldung und Ursache**
71	`duplicate APPT` Versuch, einen doppelten Termin einzugeben. Termine müssen sich in mindestens einem Zeichen unterscheiden.
72	`day/date mismatch` Wochentag und Datum stimmen nicht überein.
73	`bad day field` Falsche Buchstabierung des Wochentags.
74	`bad date field` Unzulässiger Parameter für Monat, Tag oder Jahr. Kann durch ein unerlaubtes Zeichen in den Feldern `Mo`, `Dy` oder `Yr` verursacht werden.
75	`bad time field` Unzulässiger Parameter für Stunde oder Minute. Kann durch ein unerlaubtes Zeichen in den Feldern `Hr`, `Mn` oder `AM` verursacht werden.
76	`bad rep field` • Unzulässiger Parameter oder nicht erlaubtes Zeichen in der `Rept` (*repeat*) Maske. • Drücken einer Systemtaste, während die `Rept` Maske in der Anzeige steht.
77	`bad alarm spec` • Unzulässige Alarmart. Muß eine Ziffer von `0` bis `9` sein. • Unzulässige Terminart. Muß ein `N`, `A` oder `R` sein.

Syntaxfehler (78 bis 91)	
Nummer	**Meldung und Ursache**
78	`syntax` • Unzulässige Zwischenräume oder Zeichen in der Zeile. Der Cursor steht auf dem Zeichen, wo der Fehler gefunden wurde. • Bei `TRANSFORM INTO BASIC` konnte eine Zeile nicht interpretiert werden. Der Fehler wird nach der Umformung des gesamten Files gemeldet; nicht interpretierte Zeilen werden in Programmkommentare, die mit `! ?` beginnen, umgeformt.
79	`; expected` Fehlendes Semikolon zwischen Parametern.
80	`) expected` Fehlende schließende Klammer in einem Ausdruck.
81	`comma expected` Fehlendes Komma zwischen Parametern.
82	`string expected` Nicht gelungener Versuch, einen Ausdruck als String zu interpretieren.
83	`missing TO` Das Schlüsselwort `TO` ohne Zwischenräume muß in den Befehl eingefügt werden.
84	`extra characters` Überflüssige Zeichen am Zeilenende. Kann durch falsche Eingabe einer Instruktion entstanden sein.

Syntaxfehler (78 bis 91) (*Fortsetzung*)	
Nummer	**Meldung und Ursache**
85	`expr too big` Der Ausdruck ist zu lang, um vom HP-75 ausgewertet zu werden. Kann durch zu viele verschachtelte Klammern oder zu viele Operationen im Ausdruck entstanden sein.
86	`illegal context` Nach `THEN`, `ELSE` `ON TIMER #` oder `ON ERROR` nicht erlaubte Anweisung.
87	`bad expression` • Syntaxfehler in einem Ausdruck; zum Beispiel zu viele Operatoren zwischen Operanden. • Versuch, einen LIF1 File ohne Zeilennummern in einen Text- oder BASIC-File umzuwandeln. Der File bleibt ein LIF1 File.
88	`bad statement` • Eine unzulässige Abkürzung. • Eine unvollständige Anweisung. • Versuch, eine Anweisung, die nur in Programmen erlaubt ist (zum Beispiel `GOTO`) über das Tastenfeld auszuführen.
89	`bad parameter` • Falscher Parametertyp; zum Beispiel ein Stringargument für eine Funktion mit numerischem Argument. • Eingabe von Parametern, die außerhalb des Definitionsbereichs liegen; Eingabe unzulässiger Zeichen oder zu vieler Parameter.
90	`bad line number` • Versuch einer unzulässigen Umnumerierung. Erzeugt eine Warnung; der spezifizierte Fileteil wird mit Ersatzwerten numeriert. • Versuch, mit einer `PRINT #` Anweisung eine größere Zeilennummer als 9999 zu erzeugen. Dabei entsteht ein Fehler; der Datenpointer bleibt am Fileende stehen, die Schreiboperation wird nicht ausgeführt. • Versuch, im `AUTO` Befehl eine Anfangszeilennummer größer als 9999 vorzugeben.
91	`missing parameter` Fehlen eines notwendigen Parameters in einer Funktion, einer Anweisung oder einem Befehl.

Massenspeicherfehler (92 bis 97)	
Nummer	**Meldung und Ursache**
92	`dev not mass mem` Versuch einer Massenspeicheroperation mit einer nicht zulässigen Peripherieeinheit.
93	`mass mem error` Die Massenspeichereinheit funktioniert nicht einwandfrei, möglicherweise wegen zu niedriger Batteriespannung.
94	`no medium` Kein Medium in der Massenspeichereinheit vorhanden.
95	`medium full` Das Medium in der Massenspeichereinheit kann keine weiteren Files aufnehmen. Löschen Sie einen oder mehrere Files, packen Sie das Medium oder setzen Sie ein neues Medium ein.
96	`invalid medium` Die Massenspeichereinheit kann nicht vom Medium lesen oder darauf schreiben. Möglicherweise ist das Medium nicht formatiert; initialisieren Sie das Medium. Möglicherweise ist das Medium beschädigt oder defekt, oder die Aufnahmeköpfe sind verschmutzt. Reinigen Sie die Köpfe entsprechend der Anleitung im Handbuch des Peripheriegeräts.
97	`invalid pack` Unterbrechung der `PACK` Operation. Möglicherweise muß das Medium mit `INITIALIZE` neu formatiert werden.

Alphabetische Auflistung

Meldung	Nummer
access restricted	65
arg out of range	11
ASSIGN IO needed	55
bad alarm spec	77
bad date field	74
bad day field	73
bad expression	87
bad line number	90
bad parameter	89
bad read/write	23
bad rep field	76
bad statement	88
bad time field	75
bad transmission	57
comma expected	81
CONT before RUN	31
COT or CSC inf	3
data type	33
day/date mismatch	72
dev not mass mem	92
DIM exist var	35
duplicate APPT	71
duplicate FN	37
duplicate name	64
expr too big	85
extra characters	84
file not found	62
FN calls itself	41
FN missing	39
FN parameter	40
FOR overflow	48
GOSUB overflow	49
illegal context	86
invalid DIM	36
invalid filespec	63
invalid IMAGE	52
invalid medium	96
invalid pack	97
invalid password	66
invalid subscript	27
invalid TAB	54
invalid USING	53
line too long	67
LOG(neg number)	13
LOG(0)	12
loop timeout	58
low batteries	14
mass mem error	93

Meldung	Nummer
medium full	95
missing ASSIGN#	45
missing line	32
missing NEXT	46
missing parameter	91
missing TO	83
neg^non-integer	9
no data	34
no END DEF	38
no loop response	56
no matching FOR	47
no medium	94
no value	7
not enough memory	16
not this file	20
num too large	2
num too small	1
numeric input	43
ON ERROR overflow	29
OPTION BASE	30
PRINT# to runfile	51
pulled too fast	24
pulled too slow	25
RAM is invalid	17
record overflow	28
RESTORE IO needed	60
RETURN w/o GOSUB	50
ROM missing	18
SQR(neg number)	10
string expected	82
string too long	42
syntax	78
system error	15
TAN or SEC inf	4
time adjust bad	70
too many inputs	44
too many names	59
unknown card	22
verify failed	21
workfile name?	69
write protected	19
wrong file type	68
wrong name	26
/zero	8
0^neg	5
0^0	6
>31 devices	61
) expected	80
; expected	79

Anhang F

Listings der Programme im Benutzerpaket

Inhalt

Es folgen die Listings der drei vorbespielten Programme, die mit Ihrem HP-75 geliefert werden – der Programme FINANZ, AUFGPST und NAMEN. Der Hauptteil des Handbuchs nimmt auf diese Programme Bezug, um Programmiertechniken und -anwendungen zu demonstrieren.

Das Programm FINANZ

Das Programm FINANZ wird in Abschnitt 1 auf Seite 21 besprochen.

Spezifikationen:

Eingabe: Spar- und Darlehensoption.

Eingaben bei der Darlehensoption: Anfangsbetrag, Anzahl der Perioden, Zinssatz pro Periode.

Ausgabe bei der Darlehensoption: Periodische Zahlung.

Eingaben bei der Sparoption: Anfangsbetrag, Anzahl der Perioden, Zinssatz pro Periode, periodische Einlage.

Ausgabe bei der Sparoption: Zukünftiger Betrag.

Größe:

2245 Bytes, auf vier Kartenspuren aufgenommen.

Benötigt nach der Initialisierung etwa 2990 Bytes.

```
1000 ! FINANZ - 13.07.1982
1010 DELAY 0 @ WIDTH INF
1020 E$=CHR$(27) @ H$=E$&"E" @ C0$=E$&"<" @ U$=E$&"A" @ O$="+-*/^" @ T=1
```

Programmkopf und Menu.

```
1030 DISP H$;C0$;E$&"%IC~ F I N A N Z ~" @ WAIT T
1040 DISP @ DISP "Wollen Sie "&CHR$(236)&"eihen oder "&CHR$(243)&"paren";
1050 INPUT ": "; R$ @ IF NOT LEN(R$) THEN R=0 @ GOTO 1080
1060 R=POS("LS",UPRC$(R$[1,1]))
1070 IF R#0 THEN 1100
1080 DISP @ DISP "Bitte 'l' oder 's' eingeben!" @ WAIT T
1090 DISP FNR$(4); @ GOTO 1040
1100 FOR I=1 TO 5 @ V(I)=0 @ NEXT I @ F=0
1110 OFF ERROR @ M=1 @ N=3 @ RESTORE 1160 @ DISP H$;C0$;
1120 IF R=1 THEN DISP "********** DARLEHEN **********"
1125 IF R=2 THEN DISP "********* SPARGUTHABEN *********"
1130 WAIT .5
1140 ON ERROR DISP FNR$(2*I-1);,@ GOTO 1230
```

Benutzereingaben.

```
1150 FOR I=M TO N @ READ P$
1160 DATA 'Anfangswert: ','Anzahl Perioden: ','Zinssatz/Periode: '
1170 DATA 'Einzahlung/Periode: '
1180 IF F#0 THEN 1190 ELSE V$="" @ GOTO 1230
1190 ON I GOTO 1220,1210,1200,1220
1200 V$=STR$(V(3)) @ V$=V$&" %" @ GOTO 1230
1210 V$=STR$(V(2)) @ GOTO 1230
1220 V$=STR$(V(I)) @ GOSUB 1490
1230 DISP @ DISP P$;
1240 IF F=0 AND I=3 THEN 1250 ELSE 1260
1250 PUT CHR$(136) @ V$=" %"
1260 INPUT "",V$;V$
1270 GOSUB 1550
1280 V(I)=ABS(V(I))
1290 NEXT I @ IF R=2 AND I=4 THEN N,M=4 @ GOTO 1150
1300 F=1 @ ON ERROR GOTO 1440
1310 R1=V(3)/100 @ R2=R1+1 @ R3=R2^(-V(2))
1320 IF R=1 THEN 1360
```

Berechnung bei Sparoption.

```
1330 IF V(3)=0 THEN V(5)=V(1)+V(2)*V(4) ELSE V(5)=(V(1)+V(4)*R2/R1*(1-R3))/R3
1340 I=5 @ GOSUB 1490
1350 L$="Endwert: " @ GOSUB 1450 @ GOTO 1390
```

Berechnung bei Darlehensoption.

```
1360 IF V(3)=0 THEN V(4)=V(1)/V(2) ELSE V(4)=V(1)*R1/(1-R3)
1370 I=4 @ GOSUB 1490
1380 L$="Rueckzahlung/Periode: " @ GOSUB 1450
```

Automatisches Programmende.

```
1390 OFF ERROR
1400 ON TIMER # 1,5 GOSUB 1670
1410 ON TIMER # 2,120 DISP FNR$(NUM("I")+2*R);"Ende 'FINANZ' Programm" @ STOP
1420 IF KEY$="" THEN 1420
1430 OFF TIMER # 1 @ OFF TIMER # 2 @ GOTO 1110
```

Behandlung unzulässiger Eingaben.

```
1440 DISP @ DISP "Unzulaessige Eingbe!" @ WAIT T @ GOTO 1100
```

Anzeige der Ergebnisse.

```
1450 DISP
1460 DISP @ DISP L$;
1470 IF LEN(V$)>12 THEN DISP USING "d.dddddde" ; V(I) ELSE DISP V$
1480 RETURN
```

Rundung der Ergebnisse.

```
1490 V(I)=INT(V(I))+INT(FP(V(I))*100+.5)/100 @ V$=STR$(V(I))
1500 IF V(I)>=1.E12 THEN RETURN
1510 IF V(I)<.005 THEN V(I)=0
1520 P=POS(V$,".") @ IF P=LEN(V$)-2 AND P#0 THEN 1540
1530 IF P=0 THEN V$=V$&".00" ELSE V$=V$&"0"
1540 RETURN
```

Auswertung des numerischen Ausdrucks.

```
1550 Y=2
1560 FOR J=1 TO 5
1570 X=POS(V$[Y,LEN(V$)],O$[J,J])+1 @ IF X#1 THEN 1590
1580 NEXT J @ V(I)=VAL(V$) @ RETURN
1590 IF UPRC$(V$[X-1,X-1])="E" THEN Y=X+1 @ GOTO 1560
```

Aktivieren der Fortsetzungstasten.

```
1600 U=VAL(V$[1,X-1]) @ V=VAL(V$[X+1,LEN(V$)])
1610 ON J GOTO 1620,1630,1640,1650,1660
1620 V(I)=U+V @ RETURN
1630 V(I)=U-V @ RETURN
1640 V(I)=U*V @ RETURN
1650 V(I)=U/V @ RETURN
1660 V(I)=U^V @ RETURN
1670 DISP @ DISP "(Fortsetzung mit bel. Taste)";U$;U$;U$;
1680 FOR W=1 TO 300
1690 IF KEY$#"" THEN POP @ GOTO 1430
1700 NEXT W
1710 GOSUB 1460
1720 RETURN
```

Auswahl einer Zeile zur Videoanzeige.

```
1730 DEF FNR$(R) = CHR$(27)&"%@"&CHR$(R)&CHR$(27)&"J"
```

Das Programm AUFGPST

Das Programm AUFGPST wird in Abschnitt 11 auf Seite 170 diskutiert.

Spezifikationen:

Eingaben: Anzahl der Spieler, Schwierigkeitsgrad, Initialen des/der Spieler(s).

Ausgaben: Selbstdokumentierende Einführung, Tonsignale, Positionierung des Cursors, derzeitiger Spielstand, Bonusanzeige, Endspielstand, Höchstpunktezahl.

Sondertasten: ←, →, Leertaste.

Größe:

4760 Bytes, auf acht Kartenspuren aufgenommen.

Benötigt nach der Initialisierung etwa 5714 Bytes.

```
1000 ! AUFGEPASST - 15.7.1982
1010 DELAY 0 @ WIDTH INF
1020 INTEGER L9,A9,H,N9,B1,I1,D1,E1,B9,C5,X5,Y5,Z,U,A,L,V,E,R7,R1,L1
1030 INTEGER A(12),B(12),C(2),S(2),W(2)
1040 DIM A$[26],B$[12],T$[26],Z$[26],C$[1],E1$[1],H1$[2],C1$[2],W5$[4]
1050 DIM E5$[4],P1$[1],P2$[1],R1$[2],E$[5],Y$[1]
1060 E1$=CHR$(27) @ H1$=E1$&"E" @ C1$=E1$&"<" @ W5$="["&E1$&"C"&"]"
1070 E5$=" "&E1$&"C"&" " @ P1$,P2$="" @ I$="     "
1080 T2=1.5 @ T3=.2 @ T5=.6 @ T6=0
1090 N9=6 @ B1=100 @ I1=20 @ D1=25 @ E1=40
1100 N=2*N9 @ L9,A9,H=0
```

Programmeinführung.

```
1110 DISP H1$;C1$ @ A$="AUFGEPASST !" @ Y5=0
1120 FOR I=1 TO 12
1130 FOR X5=26 TO I+9 STEP -1
1140 GOSUB 2430 @ DISP A$[I,I];" ";
1150 NEXT X5
1160 NEXT I @ WAIT .1 @ X5=10
1170 FOR I=1 TO 5
1180 GOSUB 2430 @ DISP "             "; @ GOSUB 2430 @ DISP "AUFGEPASST !";
1190 NEXT I @ WAIT 1.5 @ DISP
1200 IF Y5#0 THEN DISP H1$;C1$ @ DISP
1210 FOR I=1 TO N @ B(I),A(I)=0 @ NEXT I @ W(1),W(2)=0 @ RANDOMIZE
1220 DISP "Spiel fuer "&CHR$(177)&" oder "&CHR$(178)&" Personen";
1230 INPUT ": ",P1$; R$ @ R1$="12" @ GOSUB 2360 @ A=R1
1240 IF A=0 THEN 1220 ELSE P1$=R$
1250 DISP CHR$(236)&"eichtes oder "&CHR$(243)&"chweres Spiel";
1260 INPUT ": ",P2$; R$ @ R1$="LS" @ GOSUB 2360 @ L=R1
1270 IF L=0 THEN 1250 ELSE P2$=CHR$(NUM(R$)+32)
1280 IF L#L9 OR A#A9 THEN H=0
1290 IF A#A9 THEN I$="     "
1300 L9=L @ A9=A
1310 IF L=1 THEN A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ" @ T8=T5 @ GOTO 1330
1320 A$="<>/\{}()" @ T8=T6
1330 B$="ABCDEFGHIJKL" @ B$=B$[1,N]
1340 ON A GOSUB 1850,1900
1350 DISP M$ @ WAIT .3 @ GOSUB 2100
```

Anzeige der verborgenen Zeichen.

```
1360 Y5=8 @ X5=0 @ GOSUB 2430 @ D$=""
1370 FOR I=1 TO N @ D$=D$&" #" @ NEXT I @ D$=D$&"  Start" @ DISP D$; @ WAIT T2
1380 FOR I=1 TO N @ X5=2*I-1
1390 GOSUB 2430 @ GOSUB 2420 @ DISP CHR$(A(I)); @ WAIT T3
1400 GOSUB 2430 @ GOSUB 2410 @ DISP "#"; @ WAIT .2
1410 NEXT I
```

Spielbeginn.

```
1420 F=IP(2*RND+1) @ V=0
1430 IF A=1 THEN F=1
1440 E$=STR$(S(F))&"  " @ E$=E$[1,3] @ IF V=N9-1 THEN W(F)=0
1450 IF W(F)=1 THEN E=E1 ELSE E=I1
1460 X5=2*N+1 @ GOSUB 2430 @ WAIT .5 @ DISP I$[2*F-1,2*F]&":"&E$;
1470 BEEP 220+(F-1)*109,.1 @ IF W(F)=1 THEN DISP "*";ELSE DISP " ";
1480 X5=0 @ GOSUB 2430 @ DISP W5$; @ C(1)=1 @ C(2)=0 @ R$=KEY$
1490 FOR J=1 TO 2
```

Abfragen des Tastenfelds.

```
1500 R$=KEY$ @ IF R$="" THEN 1500 ELSE R7=NUM(R$)
1510 IF R7=32 THEN 1540
1520 IF R7<134 OR R7>135 THEN 1500
1530 ON R7-133 GOSUB 2450,2490 @ GOTO 1500
```

Treffen der Auswahl.

```
1540 C(J)=(X5+1)/2
1550 IF B(C(J))=1 THEN GOSUB 2400 @ GOTO 1500
1560 IF C(1)=C(2) THEN GOSUB 2400 @ GOTO 1500
1570 B(C(J))=1 @ X5=X5+1 @ GOSUB 2430 @ GOSUB 2420
1580 DISP CHR$(A(C(J))+128); @ X5=X5-1
1590 NEXT J
1600 IF A(C(1))#A(C(2)) THEN 1640
```

Abfrage auf Treffer.

```
1610 WAIT .2 @ BEEP 261,.05 @ BEEP 1046,.08
1620 S(F)=S(F)+E @ V=V+1 @ W(F)=1 @ GOSUB 1970
1630 IF V=N9 THEN GOSUB 2010 @ GOTO 1770 ELSE Q=1 @ GOTO 1670
```

Kein Treffer.

```
1640 S(F)=S(F)-D1 @ W(F)=0 @ GOSUB 1970
1650 B(C(1)),B(C(2)),Q=0
1660 IF S(F)=0 AND A=1 THEN 1750
```

Restaurieren der Anzeige.

```
1670 FOR I=1 TO 2
1680 X5=2*C(I)-2 @ GOSUB 2430
1690 IF Q=0 THEN GOSUB 2410 @ DISP " # ";ELSE DISP " ";CHR$(A(C(I)));" ";
1700 NEXT I
```

Auswerten der Spielergebnisse.

```
1710 IF A=1 THEN 1440
1720 IF W(F)=1 THEN 1440
1730 IF F=1 THEN F=2 ELSE F=1
1740 GOTO 1440
1750 WAIT .3 @ BEEP 41,1 @ DISP @ DISP
1760 DISP "Sie haben verloren, ";I$[2*F-1,2*F];" ..." @ WAIT T2 @ H$="R"
1770 DISP "Punktstand   ";I$[1,2];": ";STR$(S(1));
1780 IF A=2 THEN DISP "  ";I$[3,4];": ";STR$(S(2)) ELSE DISP
1790 WAIT T2 @ DISP H$;"ekord ist ";STR$(H);"." @ WAIT T2
1800 DISP "Noch ein Spiel? (";CHR$(234);"a/";CHR$(238)&"ein)";
1810 INPUT ": ","j"; R$ @ R1$="JN" @ GOSUB 2360
1820 ON R1+1 GOTO 1800,1200,1830
1830 DISP "Bis zum naechsten Mal ..."
1840 END
```

Spielaufbau für einen Spieler.

```
1850 S(1)=B1 @ S(2)=0
1860 INPUT "Ihre Initialen? ",I$[1,2];I$
1870 IF LEN(I$)<>2 THEN GOSUB 2320 @ GOTO 1860
1880 I$=UPRC$(I$) @ M$="Es geht los, "&I$&".  Aufgepasst!"
1890 RETURN
```

Anzeige des Punktstands.

```
1970 IF S(F)<0 THEN S(F)=0
1980 X5=2*N+4 @ GOSUB 2430 @ DISP "    ";
1990 GOSUB 2430 @ DISP STR$(S(F)); @ WAIT T8
2000 RETURN
```

Bestimmen des Siegers.

```
2010 WAIT T2 @ BEEP 1046,.05 @ BEEP 261,.08 @ DISP
2020 H$="R" @ Z=0 @ U=1 @ DISP
2030 IF S(1)>S(2) THEN 2060
2040 IF S(2)>S(1) THEN U=2 @ GOTO 2060
2050 IF A=2 THEN DISP "Spiel endet unentschieden." @ WAIT T2 @ Z=1
2060 IF S(U)>H THEN H=S(U) @ H$="Neuer R"
2070 IF Z=1 THEN RETURN
2080 DISP "Meinen Glueckwunsch, ";I$[2*U-1,2*U];"!" @ WAIT T2
2090 RETURN
```

Auswahl der Zeichenpaare.

```
2100 Z$=A$
2110 FOR I=1 TO N/2
2120 GOSUB 2210 @ GOSUB 2240
2130 A$=Z$ @ Y$=C$ @ Z$=B$
2140 FOR K=1 TO 2
2150 GOSUB 2210 @ GOSUB 2240
2160 A(NUM(C$)-64)=NUM(Y$)
2170 NEXT K
2180 B$=Z$ @ Z$=A$
2190 NEXT I
2200 RETURN
```

Zeichenauswahl auf Zufallsbasis.

```
2210 L1=LEN(Z$) @ X=IP(L1*RND+1)
2220 C$=Z$[X,X]
2230 RETURN
```

Verkürzen des Auswahlstrings.

```
2240 T$=""
2250 FOR J=1 TO L1 @ IF Z$[J,J]=C$ THEN 2270
2260 T$=T$&Z$[J,J]
2270 NEXT J @ Z$=T$
2280 RETURN
```

Behandlung unzulässiger Eingaben.

```
2290 GOSUB 2400 @ GOSUB 2530
2300 DISP "Bitte Eingabe wiederholen!" @ WAIT T2 @ GOSUB 2530
2310 RETURN
```

Behandlung unzulässiger Initialen.

```
2320 GOSUB 2400 @ GOSUB 2530
2330 DISP "Bitte nur zwei Initialen!" @ I$[2*A-1,2*A]="  "
2340 WAIT T2 @ GOSUB 2530
2350 RETURN
```

Abprüfen auf zulässige Eingaben.

```
2360 IF NOT LEN(R$) THEN R1=0 @ GOTO 2380
2370 R$=UPRC$(R$[1,1]) @ R1=POS(R1$,R$)
2380 IF R1<1 THEN GOSUB 2290
2390 RETURN
```

Tonsignal.

```
2400 BEEP 110,.05 @ BEEP 110,.05 @ RETURN
2410 BEEP 3000,.002 @ RETURN
2420 BEEP 6000,.002 @ RETURN
```

Positionieren des Cursors.

```
2430 DISP E1$;"%";CHR$(X5);CHR$(Y5);
2440 RETURN
```

Bewegen des Cursors nach links.

```
2450 IF X5=0 THEN GOSUB 2430 @ DISP E5$; @ X5=2*N+2 @ GOTO 2470
2460 X5=X5+2 @ GOSUB 2430 @ DISP " ";
2470 X5=X5-4 @ GOSUB 2430 @ DISP W5$;
2480 RETURN
```

Bewegen des Cursors nach rechts.

```
2490 IF X5=2*N-2 THEN GOSUB 2430 @ DISP E5$; @ X5=-2 @ GOTO 2510
2500 GOSUB 2430 @ DISP " ";
2510 X5=X5+2 @ GOSUB 2430 @ DISP W5$;
2520 RETURN
```

Löschen einer angezeigten Zeile.

```
2530 DISP E1$;"A";CHR$(13);E1$;"J";
2540 RETURN
2550 ! Spiel kann durch Modifikation der Zeilen 1080&1090 modifiziert werden.
```

Das Programm NAMEN

Das Programm NAMEN wird in Abschnitt 14 auf Seite 213 diskutiert.

Spezifikationen:

Eingaben: Filename des Textdatenfiles und die drei Felder des Datensatzes: Nachname, Vorname und Notiz.

Ausgabe: Alphabetische Liste in einen Datenfile.

Sondertasten: [+], [↑], [↓], [SHIFT] [↑], [SHIFT] [↓], [←], [→], [SHIFT] [←], [SHIFT] [→], [-], [=], [/], [.], [A]-[Z]

Größe:

2491 Bytes, auf vier Kartenspuren aufgenommen.

Benötigt nach der Initialisierung etwa 3054 Bytes.

```
1000 ! NAMEN - 15.7.1982
1010 DIM L$[110]
1020 PWIDTH INF @ WIDTH INF
1030 INPUT "Filename? ";A$
1040 ASSIGN # 1 TO A$
1050 N=0
```

Berechnen der Anzahl von Zeilen.

```
1060 ON ERROR GOTO 1080
1070 READ # 1 ; LO$,FO$,NO$ @ N=N+1 @ GOTO 1070
1080 OFF ERROR
```

Initialisieren der Parameter.

```
1090 M,P=0 @ L1$,F1$=""
```

Beginn der Befehls-Eingabeschleife.

```
1100 L$="Programm  << N A M E N >>" @ IF P=0 THEN 1140
1110 READ # 1,P ; LO$,FO$,NO$
1120 GOSUB 1870
1130 IF LEN(L$)>32 THEN DISP L$[1,32] @ GOTO 1150
1140 DISP L$
```

Abfrage auf Befehlstaste.

```
1150 K$=KEY$ @ IF K$="" THEN 1150
1160 K$=UPRC$(K$) @ K=NUM(K$)
1170 IF K$="+" THEN 1360
1180 IF K$="=" THEN 1550
1190 IF K=132 THEN P=P-(P>0) @ GOTO 1100
1200 IF K=133 THEN P=P+(P<N) @ GOTO 1100
1210 IF K=134 OR K=166 THEN 1130
1220 IF K=164 THEN P=P>0 @ GOTO 1100
1230 IF K=165 THEN P=N @ GOTO 1100
1240 IF K$>="A" AND K$<="Z" THEN 1610
1250 IF K$>="0" AND K$<="9" THEN 1610
1260 IF K$="*" THEN 1670
1270 IF K$="!" THEN 1580
1280 IF K$="-" THEN 1780
1290 IF K$=" " THEN P=N>0 @ GOTO 1100
1300 IF K$="/" THEN M=ABS(M-1) @ GOTO 1100
1310 IF K$="." THEN ASSIGN # 1 TO * @ END
1320 IF K<>135 AND K<>167 THEN 1150
1330 I=LEN(L$) @ IF I<33 THEN 1150
1340 IF M=1 THEN DISP L$[33] @ GOTO 1150
1350 DISP L$[I-31] @ GOTO 1150
```

Hinzufügen eines Eintrags.

```
1360 GOSUB 2000
1370 IF LEN(LO$)+LEN(FO$)>0 THEN 1390
1380 BEEP @ DISP "Fehler - kein Name !" @ GOTO 1360
1390 INPUT "Notiz: ";L$
1400 GOSUB 1940
1410 IF LEN(L$)>32 THEN L$=L$[1,32]
1420 N1$=L$
1430 IF LEN(LO$)=0 THEN LO$=" "
1440 GOSUB 2070
1450 IF L1$=" " THEN L1$=""
1460 IF F=1 THEN GOSUB 2100 @ GOTO 1460
1470 IF P>N THEN 1520
1480 FOR I=N TO P STEP -1
1490 READ # 1,I ; LO$,FO$,NO$
1500 PRINT # 1,I+1 ; LO$,FO$,NO$
1510 NEXT I
1520 PRINT # 1,P ; L1$,F1$,N1$
1530 N=N+1
1540 GOTO 1100
```

Suche nach einem Eintrag.

```
1550 GOSUB 2000
1560 GOSUB 2070
1570 GOTO 1590
```

Fortsetzen der Suche.

```
1580 GOSUB 2100
1590 IF F=0 THEN P=0
1600 GOTO 1100
```

Alphanumerische Taste.

```
1610 P=0
1620 P=P+1
1630 IF P>N THEN P=N @ GOTO 1100
1640 READ # 1,P ; LO$,FO$,NO$
1650 IF UPRC$(LO$)<K$ THEN 1620
1660 GOTO 1100
```

Drucken der Einträge.

```
1670 IF N=0 THEN 1100
1680 FOR I=1 TO N
1690 READ # 1,I ; LO$,FO$,NO$
1700 GOSUB 1870 @ IF M=0 THEN 1740
1710 K=POS(L$,"   ") @ IF K<>0 THEN L$[K,K+2]=" . " @ GOTO 1710
1720 IF L$[30,30]<>"." THEN 1740
1730 K=POS(L$,".") @ L$=L$[1,K-1]&" "&L$[K,30]&L$[32]
1740 PRINT L$
1750 NEXT I
1760 PRINT ""
1770 GOTO 1100
```

Löschen eines Eintrags.

```
1780 IF P=0 THEN 1100
1790 PRINT # 1,P ; ""
1800 IF P=N THEN P=N-1 @ GOTO 1840
1810 FOR I=P+1 TO N
1820 READ # 1,I ; LO$,FO$,NO$
1830 PRINT # 1,I-1 ; LO$,FO$,NO$ @ NEXT I
1840 PRINT # 1,N ; ""
1850 N=N-1
1860 GOTO 1100
```

Erzeugen einer Anzeigezeile.

```
1870 L$=LO$
1880 FO=LEN(FO$)
1890 IF LEN(L$)>0 AND FO>0 THEN L$=L$&","
1900 IF FO>0 THEN L$=L$&" "&FO$
1910 L$=L$&" "
1920 L$=L$&" " @ IF M=1 AND LEN(L$)<32 THEN 1920
1930 L$=L$&NO$ @ RETURN
```

Entfernen nachlaufender Leerzeichen.

```
1940 IF LEN(L$)=0 THEN RETURN
1950 IF L$[1,1]=" " THEN L$=L$[2] @ GOTO 1940
1960 I=LEN(L$)
1970 IF L$[I,I]=" " THEN I=I-1 @ GOTO 1970
1980 L$=L$[1,I]
1990 RETURN
```

Eingabe von Namen.

```
2000 INPUT "Nachname : ";L$
2010 GOSUB 1940 @ IF LEN(L$)>32 THEN L$=L$[1,32]
2020 LO$=L$
2030 INPUT "Vorname :  ";L$
2040 GOSUB 1940 @ IF LEN(L$)>32 THEN L$=L$[1,32]
2050 FO$=L$
2060 RETURN
```

Suchroutine.

```
2070 P=0
2080 L1$=LO$
2090 F1$=FO$
```

Fortsetzung der Suchroutine.

```
2100 F=0
2110 P=P+1
2120 IF P>N THEN RETURN
2130 READ # 1,P ; LO$,FO$,NO$
2140 LO$=UPRC$(LO$)
2150 FO$=UPRC$(FO$)
2160 IF L1$=" " THEN 2240
2170 IF LEN(L1$)=0 THEN 2210
2180 IF UPRC$(L1$)>LO$ THEN 2110
2190 IF UPRC$(L1$)<LO$ THEN RETURN
2200 IF LEN(F1$)>0 AND UPRC$(F1$)<FO$ THEN RETURN
2210 IF LEN(F1$)=0 THEN 2260
2220 IF UPRC$(F1$)<>FO$ THEN 2110
2230 GOTO 2260
2240 IF LEN(LO$)>0 OR UPRC$(F1$)<FO$ THEN RETURN
2250 IF UPRC$(F1$)>FO$ THEN 2110
2260 F=1
2270 RETURN
```

Anhang G

Glossar

Inhalt

Sonderzeichen

@ Ermöglicht Zeilen mit Mehrfachanweisungen, Tastendefinitionen und Terminbefehle.

! Ermöglicht Kommentare am Zeilenende von Programmzeilen.

? Zwei Sonderbedeutungen:

- die voreingestellte INPUT Eingabeaufforderung.
- bezeichnet einen unbekannten File in einem Katalogeintrag.

** Spezifiziert 24-Stunden Notation im TIME-Modus.
Spezifiziert im APPT-Modus entweder 24-Stunden Notation oder einen Termin außerhalb des Einhundertjahrbereichs um das heutige Datum.

A Drei Sonderbedeutungen:

- zeigt einen Terminfile in einem Katalogeintrag an.
- spezifiziert einen Termin, der sich nach der *Bestätigung* wiederholt.
- spezifiziert eine *absolute* Anpassung in der ADJST Maske.

B Zeigt einen BASIC-File in einem Katalogeintrag an.

E Exponent – stellt eine Potenz von 10 dar.

I Zeigt einen Austauschfile (LIF1) in einem Katalogeintrag an.

L Zeigt einen Language Extension (LEX) File in einem Katalogeintrag an.

N Zwei Sonderbedeutungen:

- Spezifiziert einen *normalen*, einmaligen Termin in der APPT-Maske.
- Spezifiziert eine *normale* Uhrenanpassung in der ADJST Maske.

P Zeigt einen Privatfile in einem Katalogeintrag an.

R Spezifiziert einen Wiederholungstermin, der sofort nach seinem Eintreffen neu eingeplant wird.

T Zeigt einen Textfile in einem Katalogeintrag an.

Begriffsdefinitionen

A

Allokation — Reservieren von Speicher für Programmvariablen, bevor das Programm ausgeführt wird. Programme werden mit den Befehlen RUN und CALL allokatisiert.

Anzeigeformat — Die Art und Weise, wie Information (etwa Zeit und Datum) im Anzeigefenster dargestellt wird.

anstehende Zeile — Die zur Editierung verfügbare Zeile des momentanen BASIC- oder Textfiles; die Zeile, die mit FET und dann RTN im EDIT-Modus zur Anzeige gebracht wird.

anstehende Rücksprungbedienung — Eine ausstehende Übertragung der Programmausführung nach der Abarbeitung eines Unterprogramms oder eines aufgerufenen Programms.

Anweisung — Allgemein als Instruktion innerhalb eines Programms, die den Verlauf der Programmausführung steuert, definiert; wird gewöhnlich in der Form von Programmzeilen eingegeben. Der Unterschied zwischen Anweisungen und Befehlen ist beim HP-75 nicht mehr deutlich, da die meisten Anweisungen und Befehle sowohl vom Tastenfeld aus oder in Programmen benutzt werden können.

Anzeigeeinheit — Ein Ausgabegerät in der HP-IL, dessen Einheitscode in einer DISPLAY IS Deklaration erscheint

Anzeigefelder — Die Felder in den TIME und APPT Anzeigen, die für bestimmte Informationen reserviert sind. Das Feld N in der APPT-Maske zum Beispiel ist für die gewünschte Terminart N, A oder R reserviert.

Anzeigezeichen — Jedes Zeichen, das nach SHIFT I/R direkt über das Tastenfeld angezeigt werden kann.

Anzeigezeile — Die derzeit in der Anzeige des HP-75 stehende Zeile, die einschließlich Cursor und Eingabeaufforderung bis zu 96 Zeichen lang sein kann. Das *Anzeigefenster* enthält bis zu 32 Zeichen gleichzeitig.

appt — Der Name des Terminfiles. Dieser Name erscheint in allen Befehlen ohne Anführungszeichen.

APPT-Modus — Der Betriebszustand des HP-75, in dem Sie Termine einrichten und Kalenderdaten prüfen können.

Argument — Die numerische oder Stringinformation, auf die eine Funktion wirkt.

ASCII — American Standard Code for Information Interchange; die Norm, die der HP-75 zur internen Darstellung seines Zeichensatzes benutzt. Jedes der 256 Zeichen des HP-75 entspricht einem Dezimalcode von 0 bis 255.

aufrufendes Programm — Ein Programm, das mit Hilfe der Anweisung CALL die Ausführung eines anderen Programms beginnt.

Austauschfile — Ein Logical Interchange Format (LIF1) File, der zum Austausch von Information zwischen dem HP-75 und anderen Computern dient.

B

BASIC — Beginner's All-Purpose Symbolic Instruction Code, die Programmiersprache des HP-75. Mit dem Schlüsselwort `BASIC` werden in `EDIT`-Befehlen neue Programmfiles erzeugt.

bedingte Verzweigung — Eine Programmverzweigung, die von einem Vergleichsergebnis abhängt.

Befehl — Gewöhnlich als Instruktion definiert, die bei der Ausführung über das Tastenfeld Programme oder den Computer direkt steuert. Der Unterschied zwischen Befehl und Anweisung ist beim HP-75 nicht mehr deutlich, da die meisten Befehle und Anweisungen vom Tastenfeld aus und im Programm benutzt werden können.

Begrenzer — Ein Zeichen (zum Beispiel `' '` oder `,`), das zur Einleitung oder Trennung von Parametern dient.

benutzerdefinierte Funktion — Eine numerische oder Stringfunktion, die in einem Programm mit den Anweisungen `DEF FN`, `LET FN` und `END FN` definiert ist.

Bestätigung eines Termins — Das Drücken von [ATTN] oder [SHIFT] [DEL], sobald ein fälliger Termin angezeigt wird.

bewerten — Den Wert eines Ausdrucks berechnen. Das Ergebnis ist immer eine numerische oder eine Stringkonstante.

Bildformatstring — Die Zeichen in einer der Anweisungen `IMAGE`, `DISP USING` oder `PRINT USING`, die die Formatierung von Anzeige- oder Druckerinformation festlegen.

boole'scher Wert — Einer der zwei Werte 0 und 1, die eine wahre oder falsche Bedingung symbolisieren.

Byte — Eine Maßeinheit des Speichers, bestehend aus acht Bits Daten oder einem Informationszeichen.

C

Cursor — Ein Blinkzeichen, das Ihnen Ihre Stellung in der Anzeigezeile zeigt. Mit dem *Ersetzungscursor* können Sie Zeichen überschreiben; mit dem *Einfügungscursor* werden Zeichen eingefügt.

D

Datenpointer — Der Mechanismus, der das als nächstes zu lesende Datenelement adressiert.

Deallokation — Rückgabe des bei der Initialisierung der Programmvariablen reservierten Speicherplatzes an das System. Verschiedene Systembefehle bedingen die Deallokation von Programmen. Siehe auch: Allokation.

Dezimalcode — Das numerische Äquivalent eines Anzeigezeichens, nimmt Werte von 0 bis 255 an.

dimensionieren — Deklarieren der Maximallänge einer Stringvariablen, der Anzahl der Elemente in einem Feld oder die Genauigkeit einer numerischen Variablen. Variablen werden mit den Anweisungen `DIM`, `REAL`, `SHORT` und `INTEGER` dimensioniert.

E

Editiertaste — Eine Taste oder Tastenkombination, die den Eingabebuffer steuert.

einfache numerische Variable — Ein Name (ein Buchstabe oder eine Kombination Buchstabe-Ziffer), der einen Speicherplatz für einen numerischen Wert repräsentiert. Eine einfache numerische Variable kann wie ihr Zahlenwert benutzt werden.

Eingabeaufforderung — Das am Zeilenanfang der Anzeigezeile stehende Symbol, das die Bereitschaft zur Aufnahme von Eingaben signalisiert. Der HP-75 hat drei Eingabeaufforderungen:

- Die BASIC-Eingabeaufforderung (>) steht bei der Editierung von BASIC-Files in der Anzeige.
- Die Texteingabeaufforderung (:) steht bei der Editierung von Textfiles in der Anzeige.
- Die INPUT Eingabeaufforderung erscheint bei der Ausführung der Anweisung INPUT. Wenn keine andere Eingabeaufforderung spezifiziert wurde, wird das Symbol ? benutzt.

Eingabebuffer — Ein 95 Bytes großer Speicherbereich, der die Tastenfeldeingaben aufnimmt. Kann mit CTL FET abgerufen werden.

Eingabehilfe — Eine Taste oder Tastenkombination, mit der Sie einen Zeichenstring eingeben können, ohne ihn eintasten zu müssen.

Einheitscode — Es werden zwei Arten von Einheitscodes unterschieden:

- Eines der beiden Schlüsselworte CARD und PCRD, die in einem Filespezifikator nach dem Doppelpunkt (:) einen Kartenfile spezifizieren.
- Eine aus einem oder zwei Zeichen bestehende Abkürzung, die ein Peripheriegerät in einem HP-IL System spezifiziert. Kann aus einem oder zwei Buchstaben, einem Buchstaben und einer Ziffer oder einer Ziffer und einem Buchstaben bestehen.

Einrichten eines Termins — Eingabe eines Termins in den appt-File.

End-Of-File Markierung — Eine für den Benutzer unsichtbare Marke, die zurFiletrennung vom Betriebssystem ans Ende jedes Files gesetzt wird.

Empfänger — Eine HP-IL Einheit, die in der Anweisung DISPLAY IS oder PRINTER IS deklariert worden ist.

Ersatzwert — Ein vom HP-75 eingesetzter Wert, wenn in einer Instruktion oder Maske ein Parameter nicht spezifiziert wurde, oder ein Wert, der vom System gestellt wird, um nach einer nicht erlaubten Operation die Programmausführung fortsetzen zu können.

Escapecode — Eine Zeichenfolge, die mit dem Escapezeichen eingeleitet wird, und als HP-IL oder Anzeigeinstruktion interpretiert werden kann.

Escape-Tastenfolge — CTL BACK.

Escapezeichen — Das mit CTL BACK oder CHR$(27) erzeugte Steuerzeichen (Dezimalcode 27).

EXACT Marke — Eine Zeitmarke, die für die Systemuhr gesetzt wird, wenn Sie im TIME-Modus exact RTN eingeben.

F

fälliger Termin — Ein Termin, dessen Uhrzeit und Datum eingetreten sind.

Feldelement — Einer der Werte in einem numerischen Feld; kann über Indices aufgerufen werden.

File — Eine Folge von Zeilen im Speicher oder auf einem Massenspeichermedium, die einen eigenen Namen besitzt und als Einheit betrachtet werden kann.

Filename — Ein String mit einem bis zu acht Zeichen, der einen File im Speicher spezifiziert. Das erste Zeichen muß ein Buchstabe oder ein Punkt sein. Die folgenden Zeichen können eine beliebige Kombination von Buchstaben und Ziffern sein. Alle Filenamen außer APPT und KEYS können durch Stringausdrücke spezifiziert und durch Leerzeichen abgeschlossen werden.

Filenummer — Eine ganze Zahl von 1 bis 9999, die in den Befehlen ASSIGN #, PRINT # und READ # einen BASIC- oder Textfile spezifiziert.

Filepointer — Ein Mechanismus im Speicher, der die anstehende Zeile im laufenden BASIC-, Text- oder Terminfile adressiert.

Filespezifikator — Ein Zeichenstring in der Form ' *Filename* [: *Einheitscode* [/ *Paßwort*]] ' , der einen BASIC-, Text-, Tasten-, Termin-, LEX oder Austauschfile benennt, der auf einem Massenspeichermedium abgelegt ist. Ein Filespezifikator kann aus einem Filenamen mit Einheitscode, einem Filenamen mit Einheitscode und Paßwort, oder nur aus dem Wort CARD bestehen. Alle Filespezifikatoren außer CARD können durch Stringausdrücke spezifiziert werden. Filespezifikatoren können mit Leerzeichen abgeschlossen werden.

Funktion — Eine eingebaute Routine, die auf null oder mehr Argumente wirkt und einen einzigen String- oder Zahlenwert zurückgibt. Eine im Programm erklärte *benutzerdefinierte* Funktion kann auf einer beliebigen Zahl von Argumenten operieren.

G

Genauigkeit — Die Anzahl signifikanter Stellen, mit der der HP-75 einen numerischen Wert speichert.

H

Hierarchie — Die vorgeschriebene Reihenfolge der Ausführung von Operationen in einem Ausdruck.

HP-IL — Hewlett-Packard Interface Loop. Ein vom HP-75 kontrolliertes Peripheriesystem.

I

I/O — Steht für *Input/Output*, die Ein- und Ausgabe von Daten über Peripherieeinheiten. Das Tastenfeld ist normalerweise die Eingabeeinheit, die Anzeige ist die Ausgabeeinheit des HP-75. Des weiteren ist auch der Kartenleser eine eingebaute I/O Einheit.

Index

- Eine Zahl, die die Zeile oder Spalte eines numerischen Feldelements angibt.
- Eine Zahl, die die Zeichenposition des Anfang oder Endes eines Teilstrings angibt.

Initialisieren eines Programms — Vorbereiten des Systems auf eine Programmausführung: Speicherzuordnung für Programmvariablen, Initialisierung der Variablen auf undefinierte Werte, Fehlerprüfung und Setzen weiterer Laufzeitzustände. Ein Programm wird initialisiert, wenn Sie im EDIT-Modus [RUN] drücken oder wenn die Befehle RUN oder CALL ausgeführt werden.

Initialisieren einer Variable	Zuweisen eines Anfangswerts auf eine Programmvariable.
Interface	Die Schaltkreise, die ein Steuergerät mit Peripherieeinheiten verbinden und deren Kommunikation ermöglichen.

K

Kartenfile	Ein BASIC-, Text-, Tasten-, Termin-, LEX oder Austauschfile, der auf Magnetkarten abgelegt ist.
Katalogeintrag	Eine Anzeigezeile mit Namen, Typ, Länge, Zeit und Datum eines Files im Speicher oder auf einem Massenspeichermedium.
KEYS	Spezifiziert den File keys. Der Name erscheint in allen Befehlen ohne Anführungszeichen.
keys File	Der Sondertextfile, der alle Tastendefinitionen enthält.

L

Language Extension File	Ein spezieller Programmfile auf Massenspeichermedien und Einsteck-ROMs, der neue Schlüsselworte definiert und die Fähigkeiten des Computers erweitert.
Laufzeitfehler	Ein Fehler, der nach der Initialisierung während der Programmausführung auftritt.

M

Massenspeicher-einheit	Eine I/O Einheit wie der Kartenleser oder das Digitalkassettenlaufwerk HP82161A, die zum Kopieren von Files von und auf Massenspeicher verwendet werden können.
Massenspeicher-medium	Eine Magnetkarte, Bandkassette oder Diskette, die Files des Computers extern und permanent speichern kann.
Mehrfachanweisungen	Eine Programmzeile mit zwei oder mehr Anweisungen oder Befehlen, die mit dem Symbol : verknüpft sind.
momentaner File	Der File, auf den zur Zeit der Filepointer zeigt, und der direkt editiert und gestartet werden kann usw.

N

Nullstring	Zeichenstring der Länge Null, spezifiziert durch '' oder "".
numerische Feldvariable	Eine geordnete Folge von Zahlen, *Elemente* genannt, die über ihren Zeilenindex oder ihre Zeilen- und Spaltenindices spezifiziert sind.
numerische Konstante	Ein fester numerischer Wert im Zahlenbereich des HP-75, zum Beispiel 3.14 oder -5E99.
numerische Funktion	Eine Operation, die aus einer bestimmten Anzahl und Typ von Argumenten im Definitionsbereich ein numerisches Ergebnis berechnet.

O

Operator	Ein Symbol, das die Werte von Ausdrücken vereinigt oder vergleicht. *Arithmetische*, *Vergleichs-* und *logische* Operatoren liefern ein numerisches Ergebnis; der *Stringoperator* & ergibt eine Stringgröße.

P

Parameter — Ein allgemeiner Ausdruck für String- und Zahlenwerte, die in Tastenfeld- oder Programminstruktionen benutzt werden.

Paßwort — Man unterscheidet zwei Typen:

- Das *System*paßwort ist die in einem LOCK Befehl spezifizierte Zeichenkombination. Die ersten acht Zeichen sind exakt wie im LOCK Befehl einzugeben, um den HP-75 nach einem Ausschalten wieder übernehmen zu können.
- Ein *Massenspeicher*paßwort ist eine Kombination aus bis zu vier Buchstaben und Ziffern, die in einem Massenspeicherspezifikator nach einem Schrägstrich (/) angegeben werden kann. Das Massenspeicherpaßwort begrenzt den Zugriff auf den jeweiligen File. Groß- und Kleinbuchstaben werden in Massenspeicherpaßworten nicht unterschieden – im Systempaßwort dagegen schon.

Peripheriegerät — Eine externe HP-IL Einheit, die vom HP-75 gesteuert wird.

Privatfile — Ein Kartenfile oder ein File in einem Einsteck-ROM, der in den Speicher eingelesen und gestartet, nicht aber untersucht, editiert oder umkopiert werden kann.

Programm — Eine Folge von Anweisungen, die eine Berechnung durchführt und die Eingabe, Verarbeitung und Ausgabe von Daten steuert. Programme können in BASIC-Files, LEX Files und ROMs gespeichert werden.

Programmpointer — Der Mechanismus, der die als nächste auszuführende Instruktion in einem BASIC Programm adressiert.

Programmvariable — Eine in einem Programm definierte und nur innerhalb dieses Programms abrufbare Variable.

Programmzeile — Eine Zeile eines BASIC-Files, die eine Zeilennummer und eine oder mehrere vollständige Instruktionen enthält.

R

RAM — Schreiblese-Speicher; die Schaltungen, die vom Benutzer erzeugte Programme, Daten, Texte, Termine, Variablenwerte und ähnliche Information speichern.

Rechnervariable — Eine numerische oder Stringvariable, die über das Tastenfeld, und nicht innerhalb eines Programmes erzeugt wurde. Programme können nicht auf Rechnervariablen zugreifen. Siehe auch: *Programmvariable*.

Rekursion — Siehe: *Rekursion*. Zur Definition einer Prozedur wird die Prozedur selbst verwandt.

ROM — Festwertspeicher, der nicht geändert oder gelöscht werden kann; das Betriebssystem des HP-75 residiert in ROM, zusätzlich sind optionale Einsteck-ROMs verfügbar.

Routine — Ein Programm, Programmsegment oder Unterprogramm, das die Ausführung eines größeren Programms unterstützt.

S

Schleife — Reihe von Anweisungen, die wiederholt ausgeführt werden, gewöhnlich bis eine spezifizierte Bedingung erfüllt ist.

Schleifenzähler Die einfache numerische Variable in einer FOR...NEXT Schleife, die die Anzahl der Schleifendurchgänge steuert.

Schlüsselwort Ein Wort, das für den HP-75 interpretierbar ist (wie ON, TO, DIM und GOTO). Schlüsselworte bilden den Kern von Programmzeilen.

Schreibmaschinentaste Eine Taste, die normalerweise einen Buchstaben, eine Ziffer oder ein anderes Symbol anzeigt.

Sofortausführungstaste Eine Taste oder Tastenkombination, die einen Befehl, eine Anweisung oder einen Ausdruck beim Drücken anzeigt und danach sofort ausführt.

Spur Eine der beiden Oberflächen einer Magnetkarte, die bis zu 650 Bytes eines einzelnen Files und die zugehörige Kataloginformation speichern kann.

Statusanzeige Eines der vier Felder im Anzeigefenster, die einen Betriebszustand des Systems anzeigen – **BATT**, **ERROR**, **APPT**, **PRGM**.

Steuerzeichen Ein besonderes Anzeigezeichen mit Dezimalcode zwischen 0 und 31 oder 127.

Stringausdruck Alle Stringkonstanten, Stringvariablen, Teilstrings und Stringfunktionen, oder jede Kombination von Stringausdrücken, die mit dem Operator & verbunden sind. Auch *Zeichenstring* oder *String* genannt.

Stringfunktion Eine vordefinierte Operation, die mit Zahlen- und Stringargumenten einen einzeln Stringwert berechnet. Einige Stringfunktionen, wie TIME$, benötigen kein Argument.

Stringkonstante Eine freie Folge von Zeichen, die in Anführungszeichen (' ' oder " ") eingeschlossen sind. Auch als Literal bezeichnet.

Stringvariable Ein Name (entweder ein Buchstabe oder eine Kombination Buchstabe-Ziffer) gefolgt von $, der einen Speicherplatz für Zeicheninformation repräsentiert. Eine Stringvariable kann gleichwertig mit der von ihr repräsentierten Stringinformation benutzt werden.

Syntax Die Regeln zur Buchstabierung von Schlüsselwörtern, Variablennamen, Operatoren, Filenamen usw., und dem Aufbau von Befehlen und Anweisungen.

Systemtasten Die sieben den Betrieb des Computers steuernden Tasten [ATTN], [TIME], [EDIT], [APPT], [FET], [RTN] und [RUN].

T

Tastendefinition Die derzeitige Funktion einer Taste oder Tastenkombination, die in der Anzeige eines Zeichenstrings oder in der Ausführung einer oder mehrerer Instruktionen bestehen kann.

Teilstring Ein Teil einer Stringvariablen, der aus null oder mehr zusammenhängenden Zeichen besteht. Ein Teilstring wird mit in eckigen Klammern ([ ]) eingeschlossenen Indices spezifiziert, die dem Namen der Stringvariablen folgen.

Terminbefehl Ein Befehl, der im Notizfeld eines Termins spezifiziert wird.

Text Eine freie Folge von Zeichen.

Texteditierung Das Bearbeiten eines Textfiles.

Textfile	Ein File mit alphanumerischer Information, die in numerierte Zeilen gegliedert ist.
TIME Befehlsfeld	Die fünf Positionen am rechten Rand der TIME Anzeige, die für die fünf Befehle des TIME Modus reserviert sind.
TIME Anzeige	Die Zeitangabe beim Drücken von [TIME].
Timernummer	Eine ganze Zahl von 0 bis 1000, die einen Systemtimer im Befehl `ON TIMER #` spezifiziert.

U

Umschalttasten	Die Tasten [SHIFT] und [CTL], die die Funktionen der Schreibmaschinen-, Editier- und Systemtasten umschalten.
unbedingte Verzweigung	Eine Verzweigung, die bei jeder Ausführung der Anweisung immer ausgeführt wird.
Unterprogramm	Ein Programmsegment, das mit `GOSUB` gestartet wird, und das die Programmkontrolle wieder zurückgibt, sobald ein `RETURN` gefunden wird.

V

Variable	Ein Name, der Zahlen- oder Stringinformation repräsentiert. Kann eine einfache numerische Variable (`A`), eine numerische Feldvariable (`A(,)`) oder eine Stringvariable (`A$`) sein.
Verarbeiten eines fälligen Termins	Anzeigen der Meldung oder Ausführung des Inhalts des Feldes `Note`. Die Verarbeitung erfolgt, wenn Sie zum Beispiel im APPT-Modus [RUN] drücken.
verschachtelte Schleife	`FOR...NEXT` Schleife, die in einer anderen Schleife enthalten ist.
verschachteltes Unterprogramm	Ein als Teil eines anderen Unterprogramms ausgeführtes Unterprogramm.
Verzweigung	Übertragung der Programmkontrolle an eine adressierte Anweisung.
Volativfile	Ein File, dessen Name mit einem Punkt beginnt. Er wird beim nächsten Ausschalten des HP-75 gelöscht.

W

Wagenrücklauf	Ein Steuerzeichen (Dezimalcode 13), das den Cursor an den Anfang der Anzeigezeile zurücksetzt.
Wagensteuerung	Bestimmte Escapecodefolgen zur Positionierung des Papiers und des Druckerkopfes in `PRINTER IS` Einheiten und des Cursors in `DISPLAY IS` Einheiten. Siehe: *Escapecode*.
Wert	Ein Wert ist für numerische und für Stringvariablen definiert: • Die aktuelle Zahl, die von einem Symbol (`5`), einer numerischen Variablen (`A`), einer Funktion (`sqr(20)`) oder einem Ausdruck repräsentiert wird. • Die aktuellen Zeichen, die von einem Stringausdruck dargestellt werden.

Z

Zahlenausdruck — Eine gültige Kombination von Konstanten, numerischen Variablen und numerischen Funktionen, die mit arithmetischen, Vergleichs- und logischen Operatoren verknüpft sind. Bei der Auswertung wird ein Zahlenausdruck auf einen einzigen konstanten Zahlenwert reduziert.

Zeichen — Alle 256 verschiedenen Buchstaben, Ziffern und Symbole, die mit der Funktion `CHR$` angezeigt werden können. Ein Teil davon kann direkt über das Tastenfeld angezeigt werden.

Zeichenstring — Eine Folge von Zeichen, die als Einheit behandelt werden kann. Siehe auch: *Stringausdruck*.

Zeileneditierung — Hinzufügen, Ändern oder Entfernen von Zeichen im Eingabebuffer.

Zeilennummer — Eine vorzeichenlose ganze Zahl von 0 bis 9999, die die Reihenfolge der Anweisungen in einem BASIC-File bzw. die Reihenfolge der Zeilen in einem Textfile festlegt.

Zeilenvorschub — Ein Steuerzeichen (Dezimalcode 10), das den HP- 75 auf eine neue Anzeigezeile stellt. Wird mit CTL J oder `CHR$(10)` erzeugt.

zulässige Anweisung — Eine BASIC Anweisung, die den Schlüsselworten `THEN`, `ELSE`, `ON ERROR` oder `ON TIMER` folgen kann. Alle Anweisungen des HP-75 außer `DATA`, `DEF FN`, `DIM`, `END DEF`, `FOR`, `IF`, `IMAGE`, `INTEGER`, `NEXT`, `OPTION BASE`, `REAL`, `ON ERROR`, `ON TIMER` und `SHORT` sind zulässig.

Anhang H

Syntaxzusammenfassung

Inhalt

Hierarchie der Operatoren

Die arithmetischen, logischen und Vergleichsoperatoren sind nach ihrer Präferenz aufgelistet:

Zuerst ausgeführt:

`( )` — Verschachtelte Klammern werden von innen nach außen ausgewertet.

Funktionen.

`^`

`NOT`

`*`, `DIV`, (oder CTL /)

`+`, `-`

`=`, `>`, `>=`, `<`, `<=`, `<>` oder `#`

Bei Ausdrücken werden Operatoren der gleichen Hierarchiestufe von links nach rechts abgearbeitet.

`AND`

`OR`, `EXOR`

Zuletzt ausgeführt.

Stringoperatoren

Stringausdrücke können mit dem Stringoperator (`&`) verkettet werden. Mit den Vergleichsoperatoren (`=`, `>`, `>=`, `<`, `<=`, `<>` und `#`) können Stringausdrücke verglichen werden; sie geben eine 1 bei wahrem und eine 0 bei falschem Vergleichsergebnis zurück. Zwei Strings werden als gleich betrachtet, wenn sie genau die gleichen Zeichen in genau der gleichen Reihenfolge enthalten. Der Ungleichheitsoperator vergleicht Strings Zeichen für Zeichen von links nach rechts, bis ein Unterschied gefunden wird.

Wenn ein String endet, bevor ein Unterschied auftritt, wird der kürzere String als der kleinere betrachtet. Wenn ein Unterschied gefunden wird, entscheiden die Dezimalcodes der beiden verschiedenen Zeichen, welcher String der kleinere ist.

Zahlengenauigkeit

Typ	Genauigkeit	Bereich
REAL	12 Stellen	$\pm 9.99999999999 \times 10^{\pm 499}$
SHORT	5 Stellen	$\pm 9.9999 \times 10^{\pm 99}$
INTEGER	5 Stellen	± 99999

Variablen

Einfache numerische Variablen (Beispiele: A1, B, C3)

Der Name besteht aus einem Buchstaben oder einem Buchstaben und einer Ziffer. Voreinstellung ist REAL.

Numerische Feldvariablen (Beispiele: A1(50,5), B(20,20), C3(10))

Der Name besteht aus einem Buchstaben oder einem Buchstaben und einer Ziffer. Ein Feldname kann auch gleichzeitig an eine einfache Variable im gleichen Programm vergeben werden, aber ein eindimensionales Feld und ein zweidimensionales Feld dürfen nicht den gleichen Namen besitzen. Indices in den Deklarationen DIM, REAL, INTEGER und SHORT dimensionieren die Länge der Zeile oder Zeile und Spalte. Die Untergrenze eines Feldindexes ist 0, falls nicht zu Beginn OPTION BASE 1 deklariert wurde. Die voreingestellte Obergrenze für Zeilen- und Spaltenindices ist 10.

In den Anweisungen PRINT # und READ # kann auf ganze Felder Bezug genommen werden, indem Sie einfach den Feldnamen mit Klammern angeben (zum Beispiel C()). Ein Komma in der Klammer spezifiziert ein zweidimensionales Feld (zum Beispiel C9(,)).

Stringvariablen (Beispiele: A1$, B$, C3$)

Der Name besteht aus einem Buchstaben oder einem Buchstaben und einer Ziffer, gefolgt von einem Dollarzeichen. Die voreingestellte Länge beträgt 32 Zeichen, falls sie nicht in einer DIM Anweisung anders spezifiziert worden ist. Strings können jede beliebige Länge zwischen null Zeichen und dem Maximum, das die Speicherkapazität erlaubt, haben. Eine Stringvariable wird durch Spezifikation ihres Namens und ihrer Länge in eckigen Klammern dimensioniert: DIM A1$[25]. Die *Länge* muß eine nichtnegative ganze Zahl sein.

Teilstrings (Beispiele: A1$[2,25], B$[5], C$[I,I])

Teilstrings werden über zwei numerische Ausdrücke, die in eckige Klammern zu setzen sind, spezifiziert. Zwei Indices, die durch ein Komma getrennt sind, spezifizieren Anfangs- und Endposition der Zeichen des Teilstrings. Ein einzelner Index spezifiziert ein Anfangszeichen – der Teilstring erstreckt sich bis zu dem Ende des Strings.

Vordefinierte BASIC Funktionen

In den folgenden Tabellen spezifizieren X und Y beliebige numerische Ausdrücke und S$ und T$ beliebige Stringausdrücke.

Numerische Funktionen

Bei Spezifikation der geeigneten Argumente (in Typ und Anzahl) gibt jede numerische Funktion eine einzelne numerische Konstante zurück.

`ABS(X)`	Absolutbetrag von `X`.
`ACOS(X)`	Arcus Cosinus von `X` im ersten oder zweiten Quadranten.
`ANGLE(X,Y)`	Arcus Tangens von `Y/X` im richtigen Quadranten. Rückgabe des Winkels zwischen der x-Achse und dem Punkt (x,y), sodaß $-\pi < \theta \leqslant \pi$.
`ASIN(X)`	Arcus Sinus von `X`, im 1. oder 4. Quadranten.
`ATN(X)`	Arcus Tangens von `X`, im 1. oder 4. Quadranten.
`CEIL(X)`	Kleinste ganze Zahl $\geqslant$ `X`.
`COS(X)`	Cosinus von `X`.
`COT(X)`	Cotangens von `X`.
`CSC(X)`	Cosecans von `X`.
`DATE`	Das Datum im Format *yyddd*, basierend auf der momentanen Systemzeit.
`DEG(X)`	Umwandlung des Werts `X` von Radiant in Altgrad.
`EPS`	Kleinste Maschinenzahl (`1.E-499`).
`ERRL`	Zeilennummer des letzten Fehlers oder der letzten Warnung.
`ERRN`	Nummer des letzten Fehlers oder der letzten Warnung.
`EXP(X)`	e^{X}.
`FLOOR(X)`	Wie `INT(X)`.
`FP(X)`	Gebrochener Anteil von `X`.
`INF`	Größte Maschinenzahl (`9.99999999999E499`)
`INT(X)`	Größte ganze Zahl $\leqslant$ `X`.
`IP(X)`	Ganzzahliger Anteil von `X`.
`LEN(S$)`	Länge des Strings `S$`.
`LOG(X)`	Natürlicher Logarithmus von `X`, `X` > 0.
`LOG10(X)`	Zehnerlogarithmus von `X`, `X` > 0.
`MAX(X,Y)`	Wenn `X` > `Y`, dann `X`, sonst `Y`.
`MEM`	Anzahl noch verfügbarer Bytes im Speicher.
`MIN(X,Y)`	Wenn `X` < `Y`, dann `X`, sonst `Y`.

MOD(X,Y)	X modulo Y: X-Y*INT(X/Y).
NUM(S$)	Dezimalcode des ersten Zeichens von S$.
PI	3.14159265359.
POS(S$,T$)	Sucht nach dem ersten Auftreten von T$ in S$. Gibt die Anfangsposition oder, falls nicht gefunden, 0 zurück.
RAD(X)	Umwandlung des Werts X von Altgrad in Radiant.
RES	Letztes angezeigtes oder ausgedrucktes Zahlenergebnis.
RMD(X,Y)	Rest von X/Y: X-Y*IP(X/Y).
RND	Die nächste Zahl R in einer Folge von Zufallszahlen; $0 \leqslant R < 1$.
SEC(X)	Sekans von X.
SGN(X)	Das Vorzeichen von X: –1, falls $X < 0$; 0, falls $X = 0$; 1, falls $X > 0$.
SIN(X)	Sinus von X.
SQR(X)	Positive Quadratwurzel von X.
TAN(X)	Tangens von X.
TIME	Anzahl der Sekunden seit Mitternacht.
VAL(S$)	Numerischer Wert eines Strings aus Ziffern, Dezimalpunkt und/oder Exponent.

Stringfunktionen

Stringfunktionen geben null oder mehr Zeichen an Information zurück.

CAT$(X)	Der Katalogeintrag des spezifizierten Files mit 32 Zeichen Länge. Files werden in der Reihenfolge ihres Eintrags im Systemkatalog numeriert. CAT(0) gibt den Katalogeintrag des momentanen EDIT-Files zurück. Bei $X < 0$ gibt CAT$(X) den Katalogeintrag des derzeit initialisierten BASIC-Files zurück, falls vorhanden.
CHR$(X)	Zeichen mit dem Dezimalcode MOD(X,256).
DATE$	Datum im Format *yy/mm/dd*.
KEY$	Anzeigezeichen der derzeit gedrückten Taste oder Tastenkombination. Falls keine Taste gedrückt ist, wird ein Nullstring zurückgegeben.
STR$(X)	Die in den Ziffern, dem Dezimalpunkt, Vorzeichen und Exponenten von X enthaltene Stringinformation.
TIME$	Zeit im Format *hh:mm:ss* in 24-Stunden Notation.
UPRC$(S$)	Wandelt die Kleinbuchstaben in S$ in Großbuchstaben um.
VER$	Ein sechsstelliger String, der die Version des Betriebssystems enthält.

Druckerfunktion

TAB(X)	Die folgende DISP oder PRINT Ausgabe wird von Spalte X an dargestellt, wobei Spalte 1 der linke Rand ist.

TIME-Modus Befehle

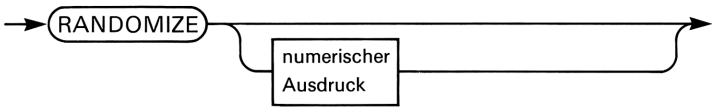
Die folgenden Befehle sind nicht programmierbar und nur im TIME-Modus ausführbar.

ADJST	Zeigt die ADJST Maske, die zum Anpassen der Ganggeschwindigkeit der Uhr dient, an.
EXACT	Setzt eine Zeitmarke zur Eichung der Uhr.
RESET	Löscht vorhergehende EXACT Marken und den derzeitigen Faktor zur Geschwindigkeitsanpassung.
SET	Zeigt die Zeitsetzmaske an, mit der die Systemuhr gestellt wird.
STATS	Zeigt die STATS Maske an, mit der die Anzeigeformate für Monat/Tag und Stunde gesetzt werden und der Jahres- oder der erweiterte Kalender im APPT-Modus spezifiziert werden.

Syntaxrichtlinien für BASIC Anweisungen und Systembefehle

In Ovalen erscheinende Elemente können in Groß- oder Kleinschrift eingegeben werden. In den Rechtecken stehen die Parameter, die in den Befehlen oder Anweisungen benutzt werden können (oder müssen).

Beispiel:



Den die Symbole verbindenden Linien kann beim Aufbau des Befehls oder der Anweisung nur in einer Richtung gefolgt werden. Ein Element (Schlüsselwort oder Parameter) ist optional, wenn ein Weg um es herum führt. Es kann also jede Kombination der Elemente benutzt werden, solange Sie den Pfeilen in der richtigen Richtung folgen. Daraus ergibt sich, daß zwei Formen des Befehls RANDOMIZE, nämlich RANDOMIZE oder RANDOMIZE *numerischer Ausdruck*, möglich sind.

Wenn Parameter in Anführungszeichen gesetzt werden müssen, können Sie wahlweise einfache (' ') oder doppelte ('' '') Anführungszeichen verwenden, allerdings immer nur paarweise. Jeder Parameter in Anführungszeichen, wie '*Filename*' und ':*Einheitscode*', kann durch einen Stringausdruck spezifiziert werden.

Die ersten zwei Buchstaben eines Schlüsselwortes dürfen nicht durch Leerzeichen getrennt sein; ansonsten dürfen Befehle und Anweisungen mit beliebig vielen eingeschlossenen Leerzeichen eingegeben werden.

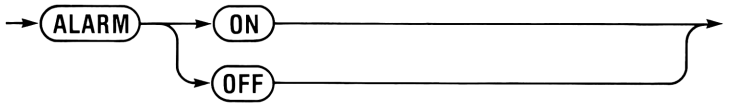
Ein ganzer Pfeil (——►), der auf das Schlüsselwort zeigt (wie im Beispiel RANDOMIZE zuvor), bedeutet:

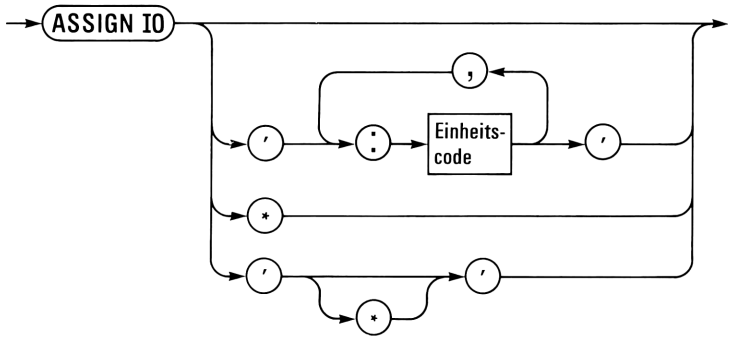
- Dem Befehl oder der Anweisung kann eine weitere Instruktion in der gleichen Programmzeile folgen, wenn sie mit dem Symbol @ verknüpft wird.
- Der Befehl oder die Anweisung kann in einer IF...THEN...ELSE Anweisung erscheinen.

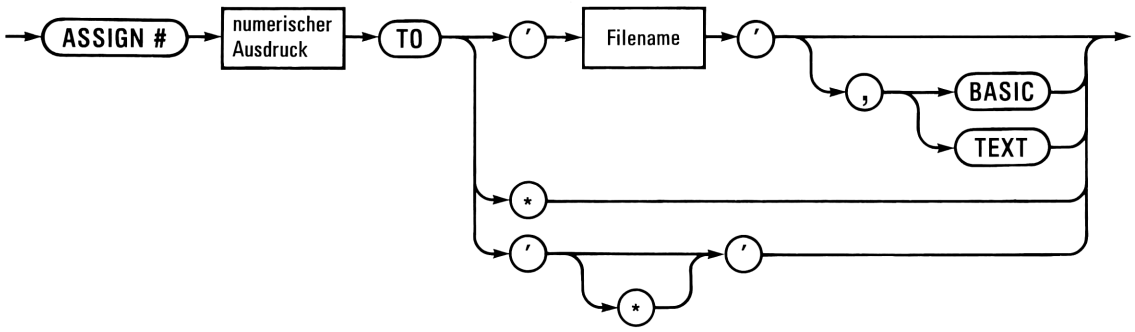
Ein auf das Schlüsselwort zeigender halbierter Pfeil (——►) bedeutet, daß die Anweisung nach THEN, ELSE, ON ERROR oder ON TIMER nicht zulässig ist. Ein fehlender Pfeil bedeutet, daß die Anweisung als erste Instruktion einer Programmzeile eingegeben werden muß.

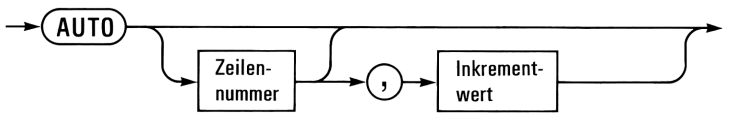
Ein von der Anweisung wegführender ganzer Pfeil bedeutet, daß in der gleichen Zeile eine weitere Anweisung folgen darf. Ein ——X bedeutet, daß keine weitere Anweisung folgen darf. Ein ——| bedeutet, daß eine Anweisung folgen darf, die aber nie ausgeführt wird, wie nach der Anweisung END.

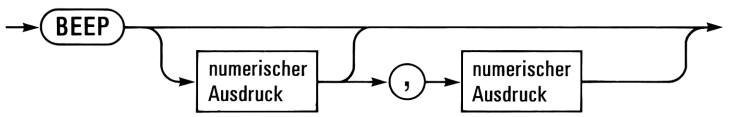
Schlagen Sie die möglichen Abkürzungen der Schlüsselworte in Anhang D «Referenztabellen» nach.

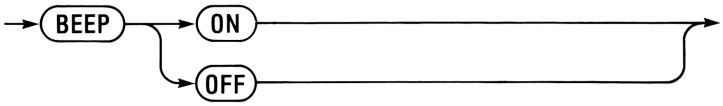


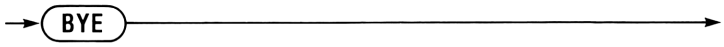


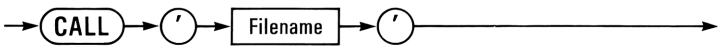


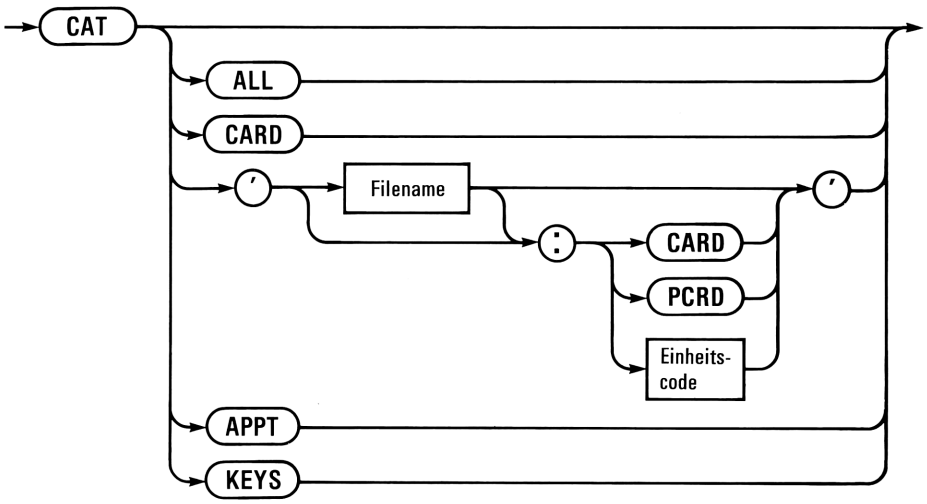


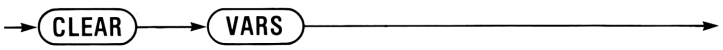


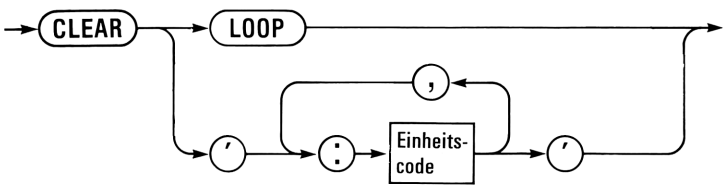

BEEP
ON
OFF

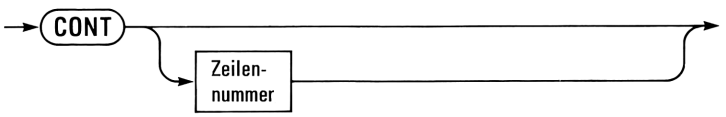

BYE

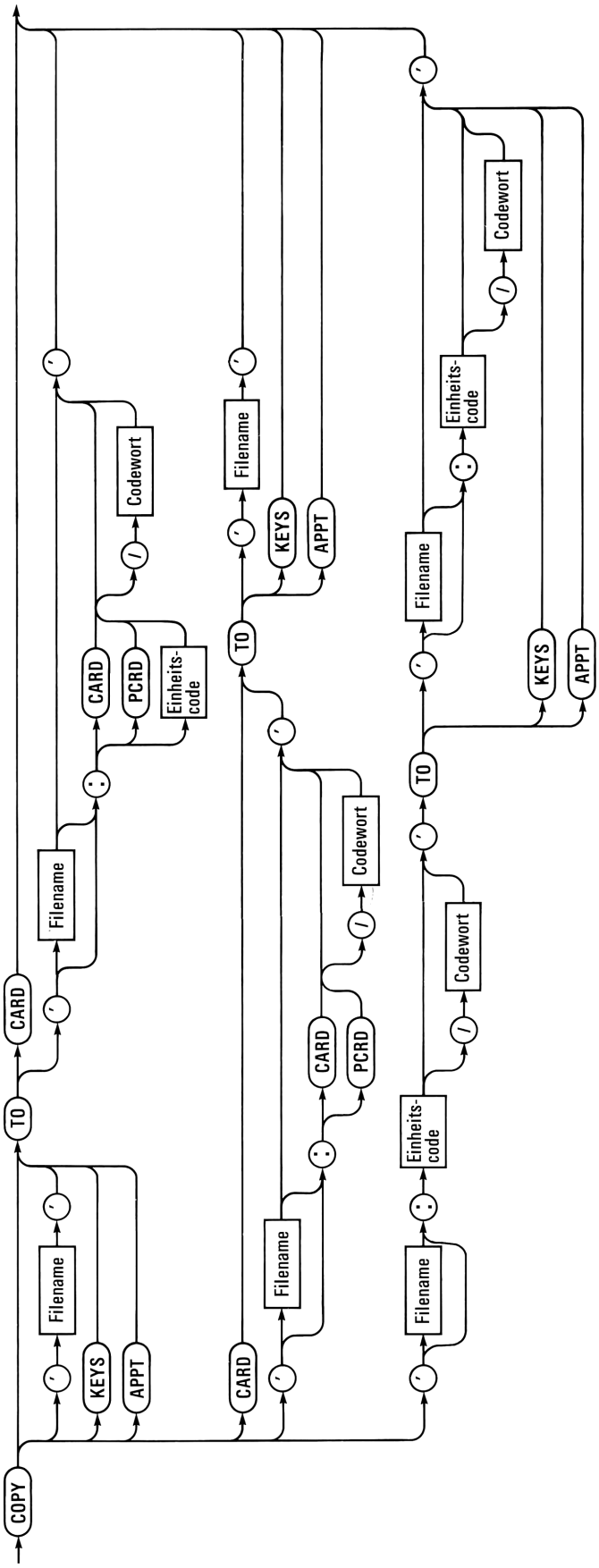

CALL
'
Filename
'

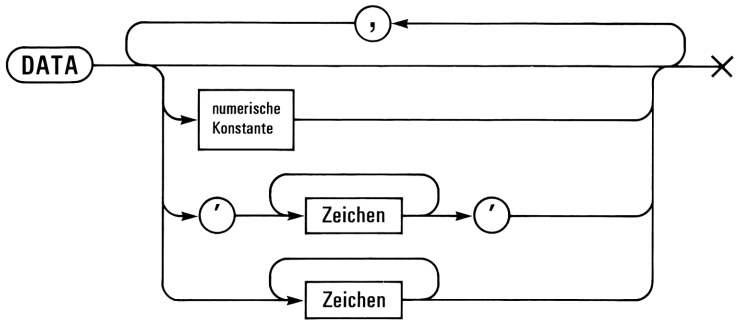

CAT
ALL
CARD
'
Filename
'
:
CARD
PCRD
Einheits-
code
APPT
KEYS

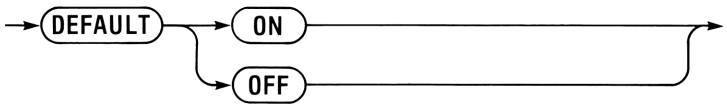

CLEAR
VARS

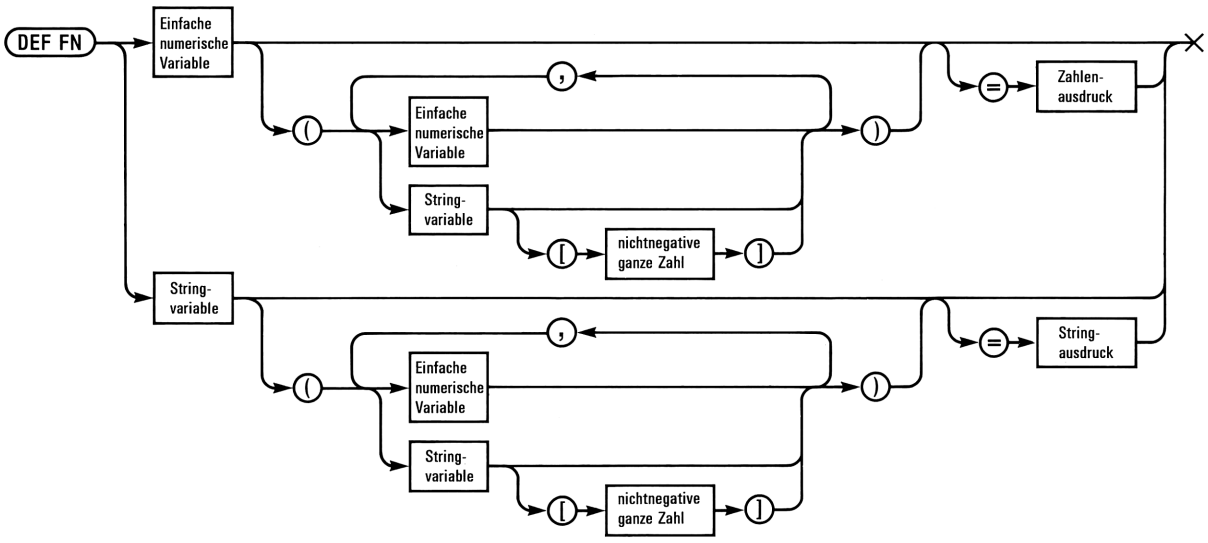

CLEAR
LOOP
,
'
:
Einheits-
code
'

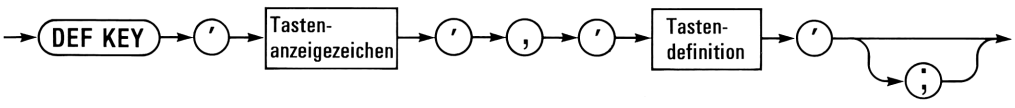

CONT
Zeilen-
nummer

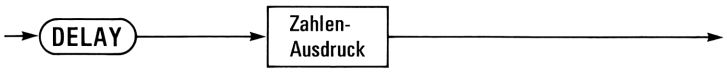

COPY
Filename
KEYS
APPT
TO
CARD
Filename
Einheits-code
Codewort
PCRD
CARD
Filename
KEYS
APPT
TO
Filename
Einheits-code
Codewort
CARD
PCRD
Filename
Einheits-code
Codewort
TO
Filename
KEYS
APPT

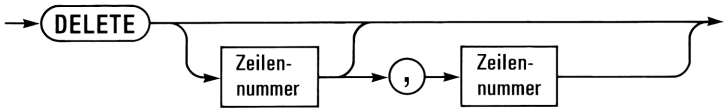

DATA
,
numerische Konstante
'
Zeichen
'
Zeichen

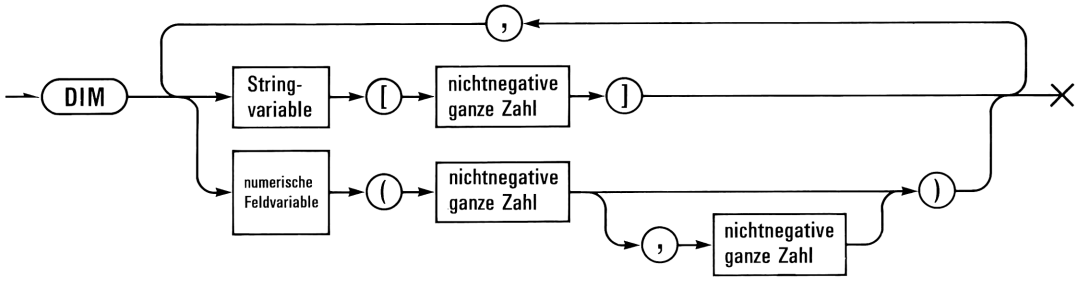

DEFAULT
ON
OFF

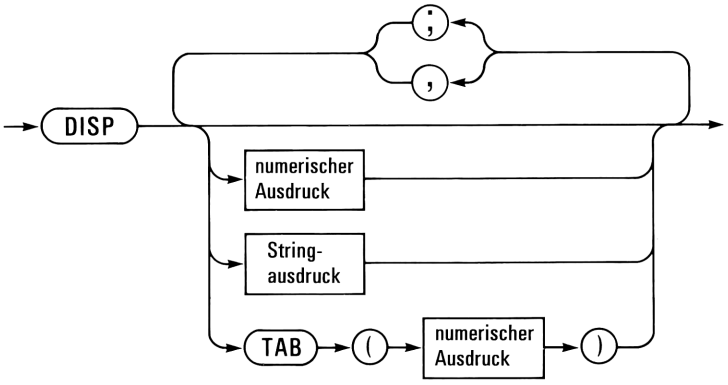

DEF FN
Einfache numerische Variable
(
,
Einfache numerische Variable
String-variable
[
nichtnegative ganze Zahl
]
)
=
Zahlen-ausdruck
String-variable
(
,
Einfache numerische Variable
String-variable
[
nichtnegative ganze Zahl
]
)
=
String-ausdruck

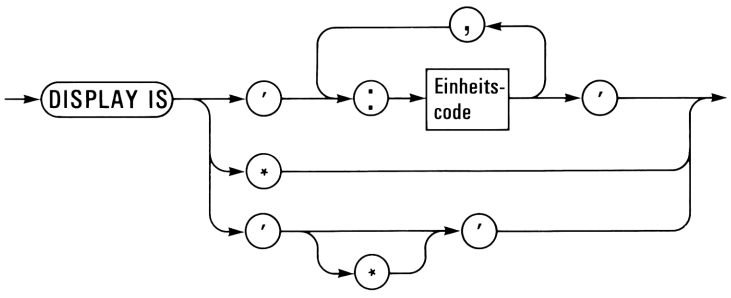

DEF KEY
'
Tasten-anzeigezeichen
'
,
'
Tasten-definition
'
;

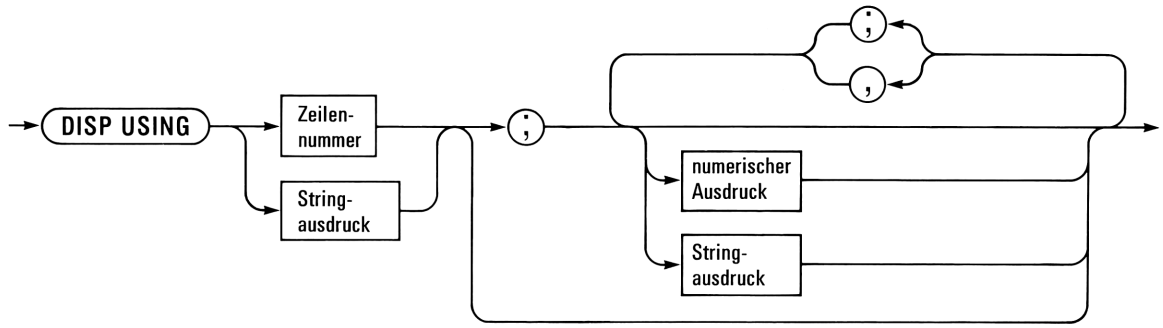

DELAY
Zahlen-Ausdruck

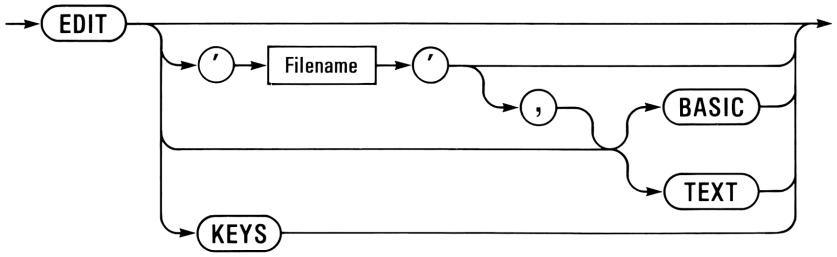

DELETE
Zeilen-
nummer
,
Zeilen-
nummer


DIM
String-
variable
[
nichtnegative
ganze Zahl
]
numerische
Feldvariable
(
nichtnegative
ganze Zahl
,
nichtnegative
ganze Zahl
)

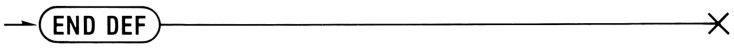

;
,
DISP
numerischer
Ausdruck
String-
ausdruck
TAB
(
numerischer
Ausdruck
)

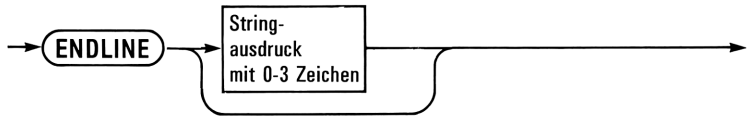

,
DISPLAY IS
'
:
Einheits-
code
'
*
'
'
*

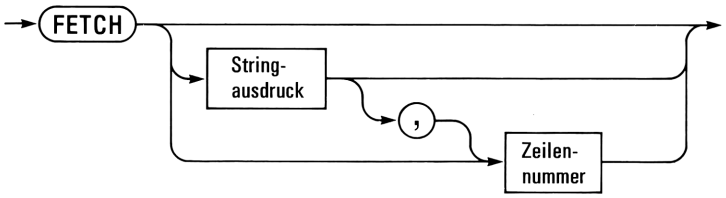

DISP USING
Zeilen-
nummer
String-
ausdruck
;
;
,
numerischer
Ausdruck
String-
ausdruck

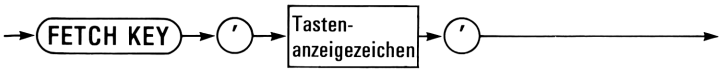

EDIT
'
Filename
'
,
BASIC
TEXT
KEYS

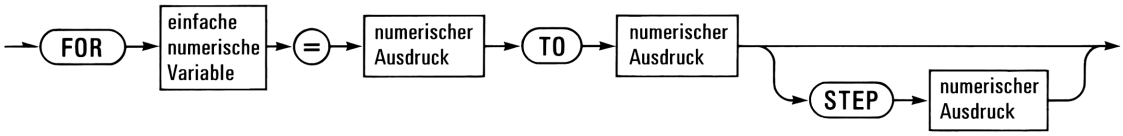

END

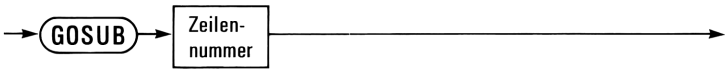

END DEF

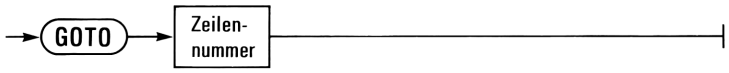

ENDLINE
String-
ausdruck
mit 0-3 Zeichen

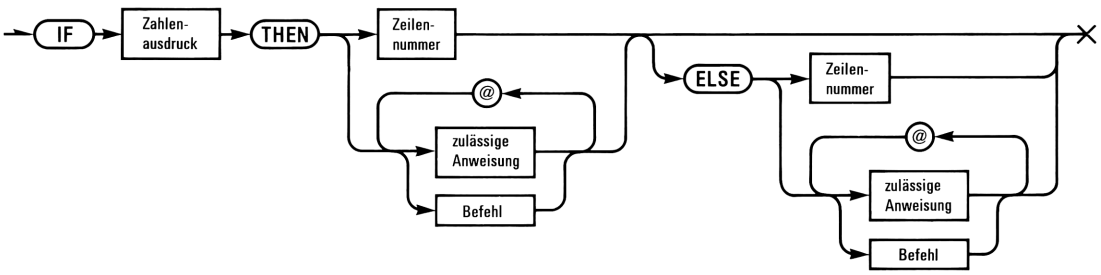

FETCH
String-
ausdruck
,
Zeilen-
nummer


FETCH KEY
'
Tasten-
anzeigezeichen
'


FOR
einfache numerische Variable
=
numerischer Ausdruck
TO
numerischer Ausdruck
STEP
numerischer Ausdruck


GOSUB
Zeilen-nummer


GOTO
Zeilen-nummer


IF
Zahlen-ausdruck
THEN
Zeilen-nummer
@
zulässige Anweisung
Befehl
ELSE
Zeilen-nummer
@
zulässige Anweisung
Befehl

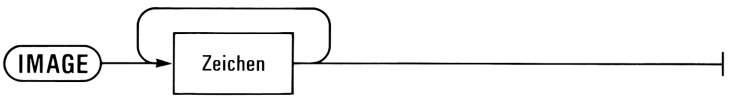
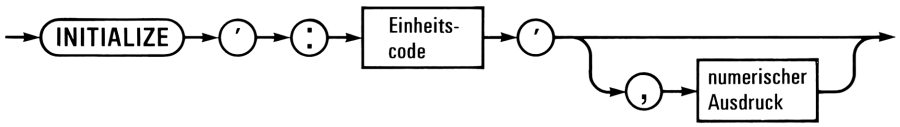
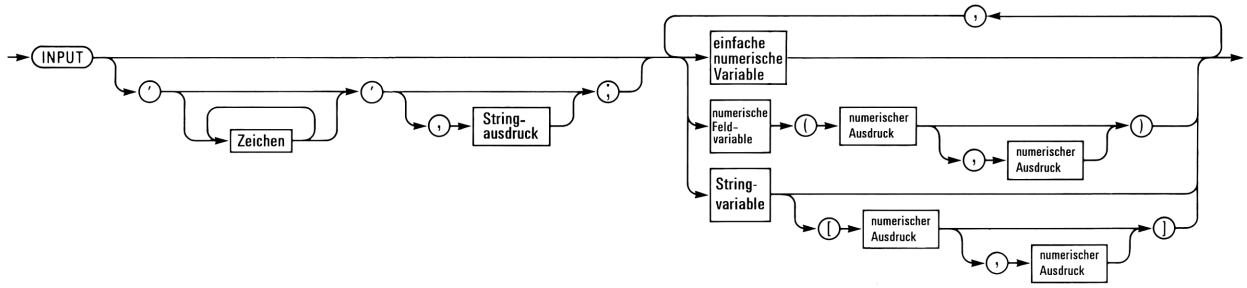
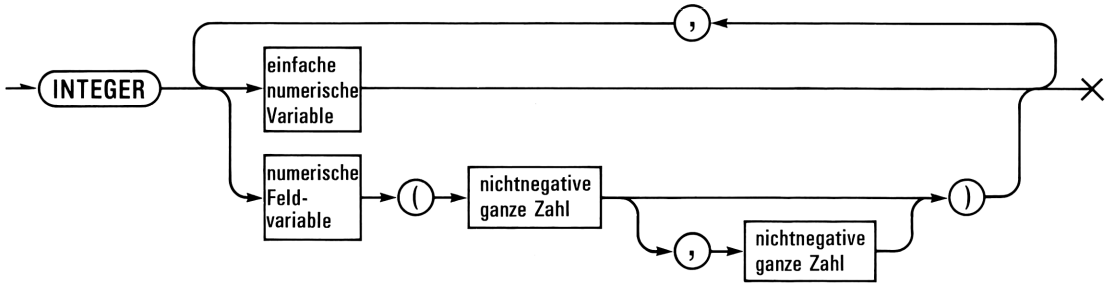
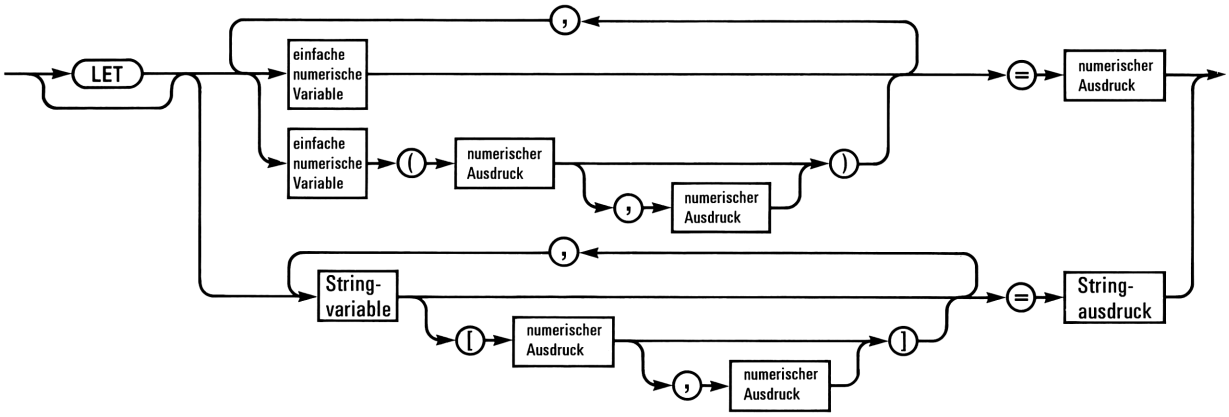


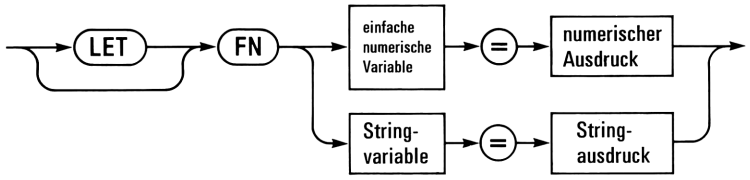
IMAGE
Zeichen

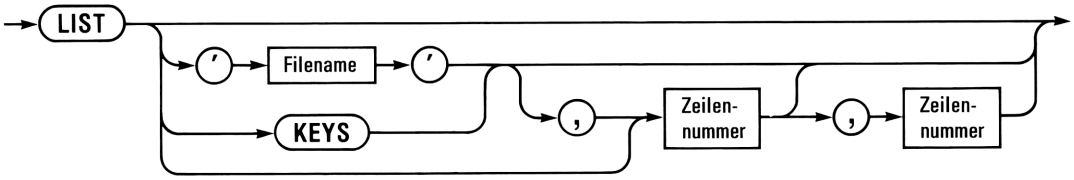

INITIALIZE
'
:
Einheits-code
'
,
numerischer Ausdruck

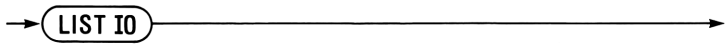

INPUT
'
Zeichen
'
,
String-ausdruck
;
,
einfache numerische Variable
numerische Feld-variable
(
numerischer Ausdruck
,
numerischer Ausdruck
)
String-variable
[
numerischer Ausdruck
,
numerischer Ausdruck
]

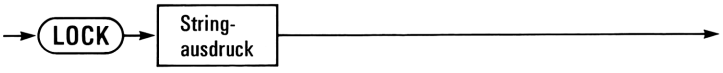

INTEGER
einfache numerische Variable
numerische Feld-variable
(
nichtnegative ganze Zahl
,
nichtnegative ganze Zahl
)
,

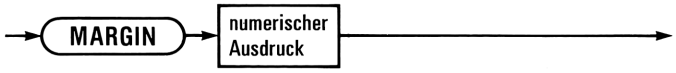

LET
einfache numerische Variable
einfache numerische Variable
(
numerischer Ausdruck
,
numerischer Ausdruck
)
=
numerischer Ausdruck
String-variable
(
numerischer Ausdruck
,
numerischer Ausdruck
)
=
String-ausdruck

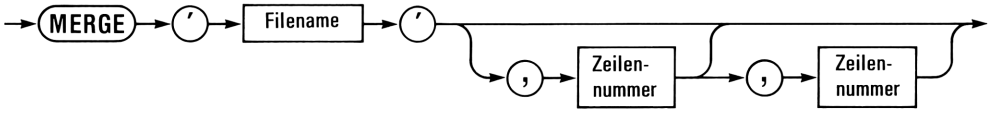

LET
FN
einfache numerische Variable
=
numerischer Ausdruck
String-variable
=
String-ausdruck

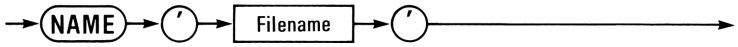

LIST
'
Filename
'
KEYS
,
Zeilen-nummer
,
Zeilen-nummer

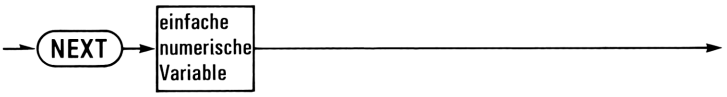

LIST IO

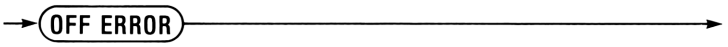

LOCK
String-ausdruck

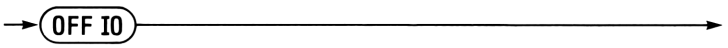

MARGIN
numerischer Ausdruck

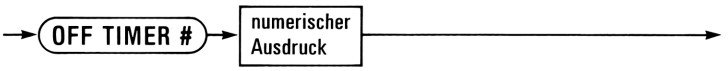

MERGE
'
Filename
'
,
Zeilen-
nummer
,
Zeilen-
nummer

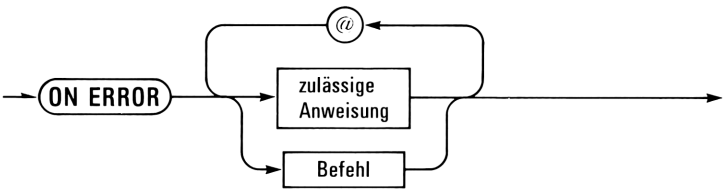

NAME
'
Filename
'

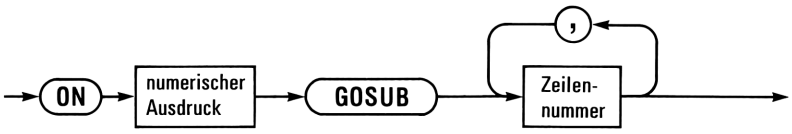

NEXT
einfache
numerische
Variable


OFF ERROR

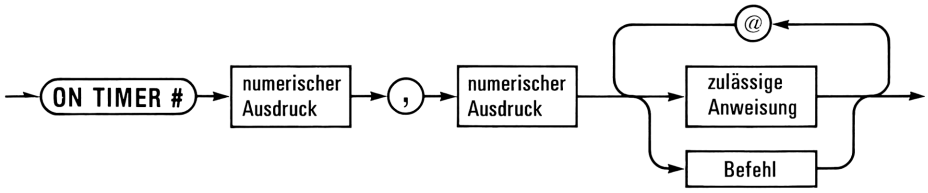

OFF IO


OFF TIMER #
numerischer
Ausdruck

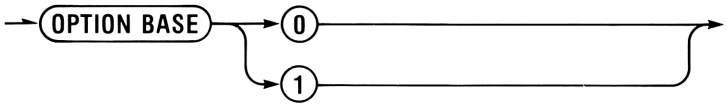

@
ON ERROR
zulässige
Anweisung
Befehl

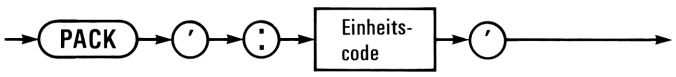

,
ON
numerischer
Ausdruck
GOSUB
Zeilen-
nummer

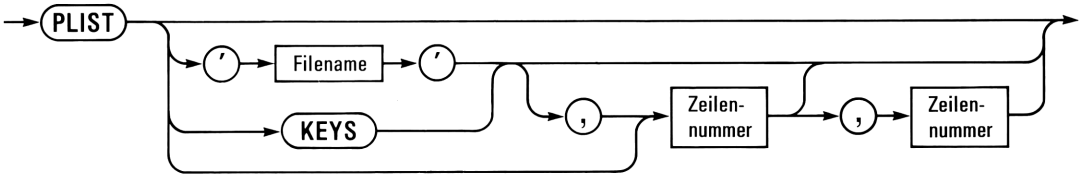

,
ON
numerischer
Ausdruck
GOTO
Zeilen-
nummer

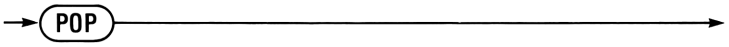

ON TIMER #
numerischer Ausdruck
,
numerischer Ausdruck
@
zulässige Anweisung
Befehl

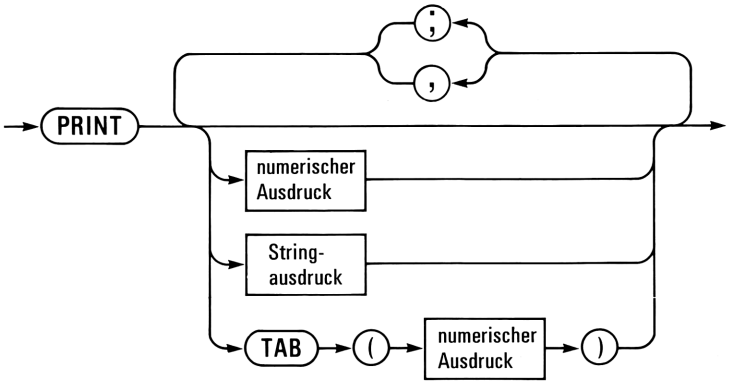

OPTION ANGLE
RADIANS
DEGREES

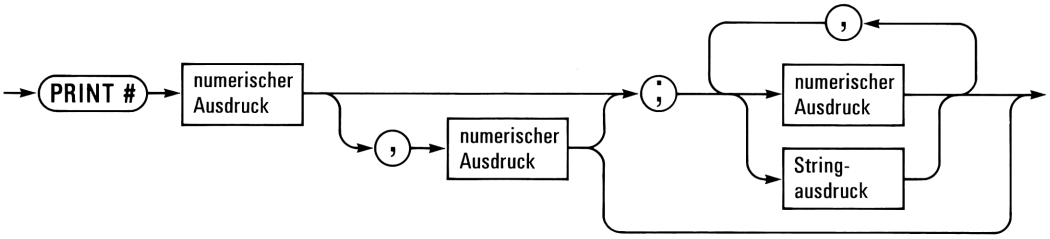

OPTION BASE
0
1

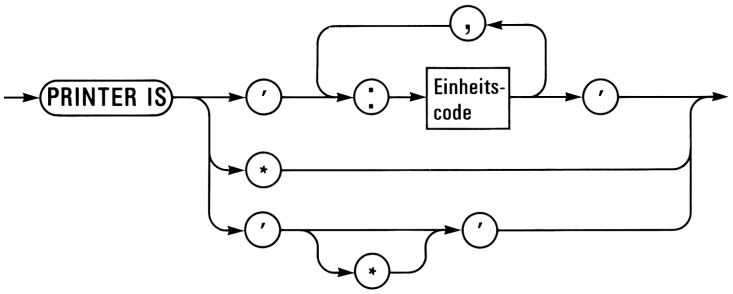

PACK
'
:
Einheits-code
'

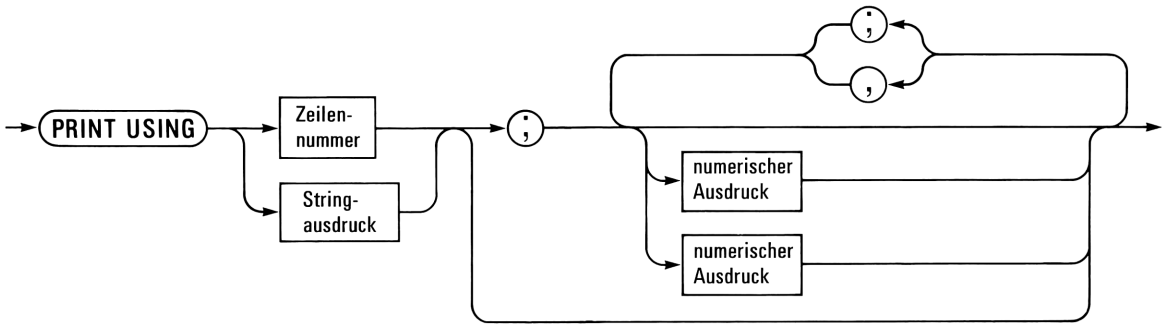

PLIST
'
Filename
'
KEYS
,
Zeilen-nummer
,
Zeilen-nummer

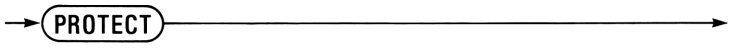

POP

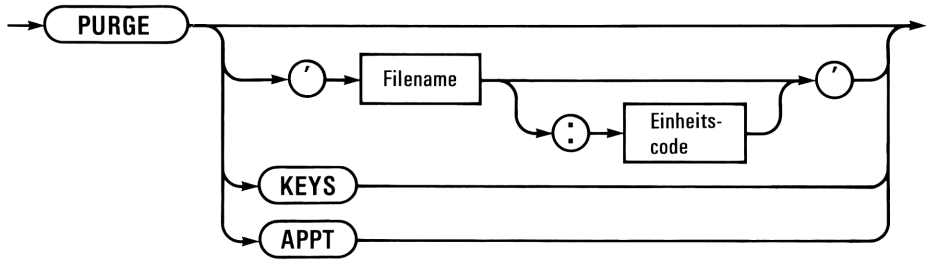

PRINT
;
,
numerischer Ausdruck
String-ausdruck
TAB
(
numerischer Ausdruck
)

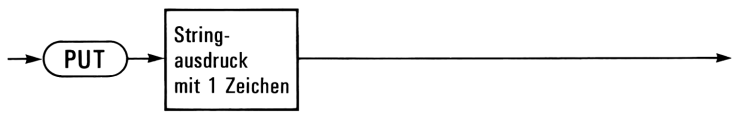

PRINT #
numerischer Ausdruck
,
numerischer Ausdruck
;
,
numerischer Ausdruck
String-ausdruck

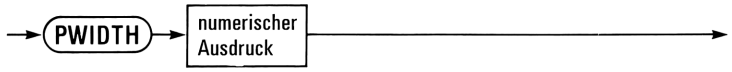

PRINTER IS
'
,
:
Einheits-code
'
*
'
'
*

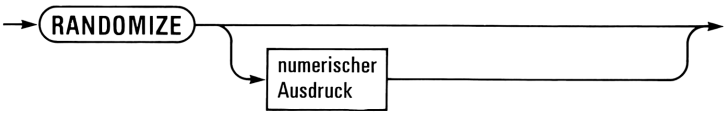

PRINT USING
Zeilen-nummer
String-ausdruck
;
;
,
numerischer Ausdruck
numerischer Ausdruck

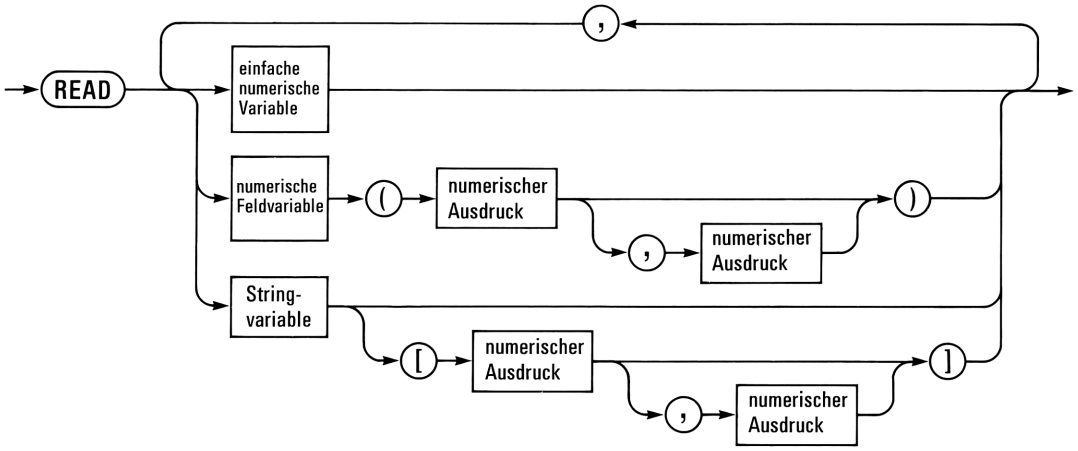

PROTECT

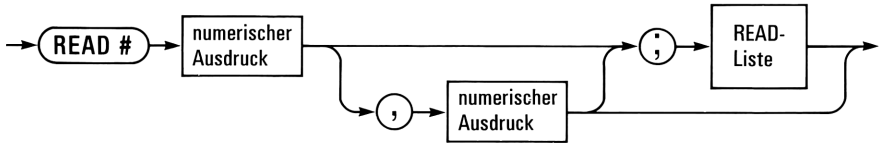

PURGE
'
Filename
:
Einheits-code
'
KEYS
APPT

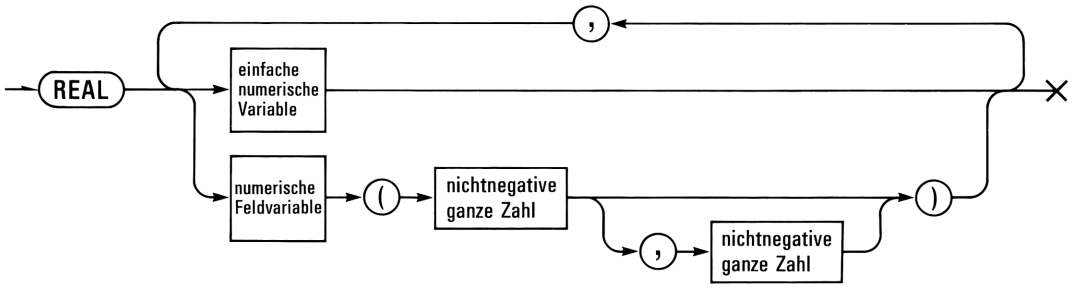

PUT
String-ausdruck mit 1 Zeichen

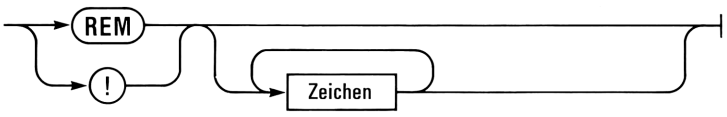

PWIDTH
numerischer Ausdruck

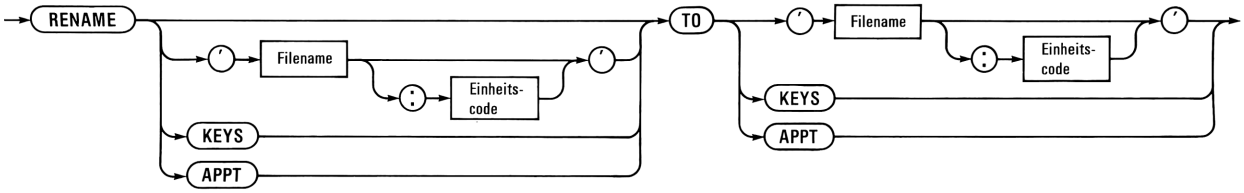

RANDOMIZE
numerischer Ausdruck

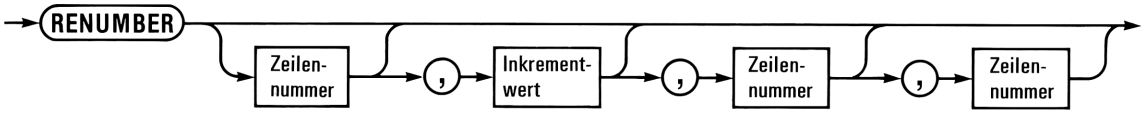

READ
,
einfache numerische Variable
numerische Feldvariable
(
numerischer Ausdruck
,
numerischer Ausdruck
)
String-variable
[
numerischer Ausdruck
,
numerischer Ausdruck
]

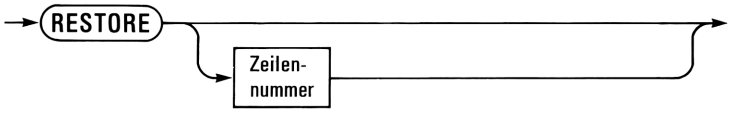

READ #
numerischer Ausdruck
,
numerischer Ausdruck
;
READ-Liste


REAL
,
einfache numerische Variable
numerische Feldvariable
(
nichtnegative ganze Zahl
,
nichtnegative ganze Zahl
)

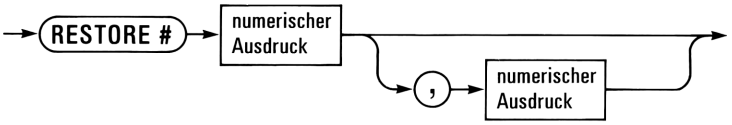

REM
!
Zeichen


RENAME
'
Filename
:
Einheits-
code
'
KEYS
APPT
TO
'
Filename
:
Einheits-
code
'
KEYS
APPT

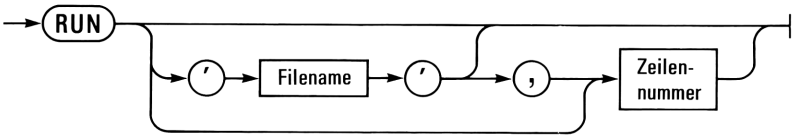

RENUMBER
Zeilen-
nummer
,
Inkrement-
wert
,
Zeilen-
nummer
,
Zeilen-
nummer

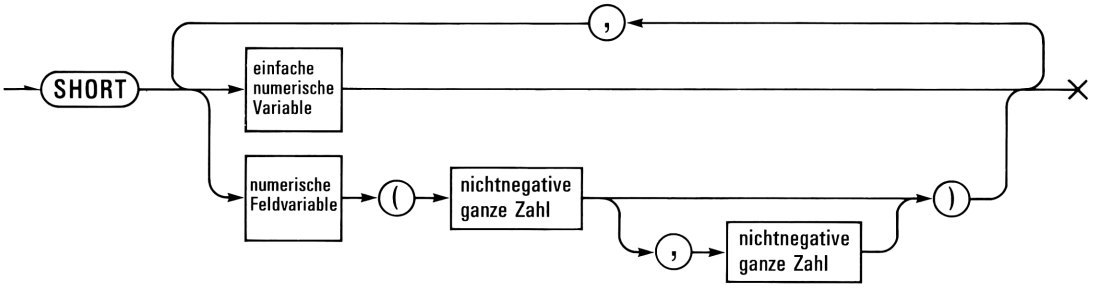

RESTORE
Zeilen-
nummer

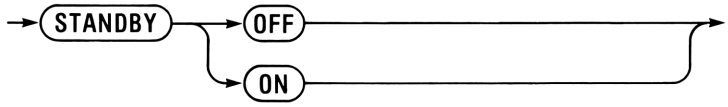

RESTORE IO

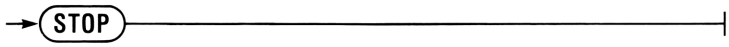

RESTORE #
numerischer
Ausdruck
,
numerischer
Ausdruck

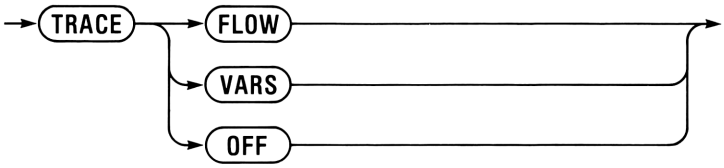

RETURN

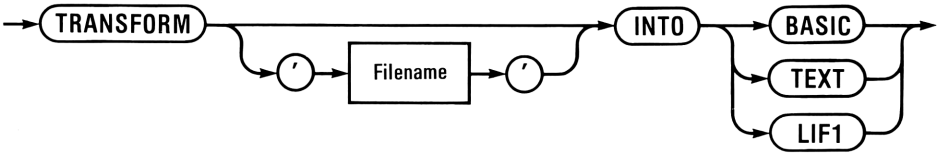

RUN
'
Filename
'
,
Zeilen-
nummer

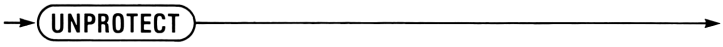

SHORT
,
einfache
numerische
Variable
numerische
Feldvariable
(
nichtnegative
ganze Zahl
,
nichtnegative
ganze Zahl
)

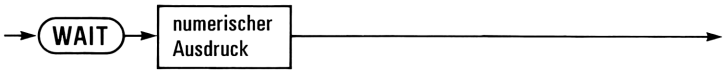

STANDBY
OFF
ON

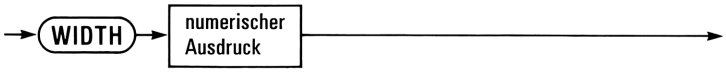

STOP


TRACE
FLOW
VARS
OFF


TRANSFORM
'
Filename
'
INTO
BASIC
TEXT
LIF1


UNPROTECT


WAIT
numerischer
Ausdruck


WIDTH
numerischer
Ausdruck

Register

Sachindex

Seitenummern in Fettdruck stehen für erste Referenzen; Seitennummern in Normaldruck verweisen auf zweite Referenzen. Zusätzlich zu diesem Sachindex finden Sie auf der Innenseite des Umschlags dieses Benutzerhandbuchs eine vollständige Zusammenstellung des Instruktionssatzes des HP-75 – Operatoren, Funktionen, Befehle und Anweisungen – mit Seitenangaben.

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Z

Index HP-75 Instruktionen

* Die zweite Seitenreferenz bezieht sich auf HP-IL Anwendungen der Befehle.

† Die zweite STATS Referenz und die EXTD Referenz beziehen sich auf den APPT Modus.

VERKAUFSNIEDERLASSUNGEN:

Hewlett-Packard GmbH:
6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1
7030 Böblingen, Herrenbergerstraße 110, Tel. (07031) 667-1
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 5971-1
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 63804-1
8028 Taufkirchen, Eschenstraße 5, Tel. (089) 6117-1
3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 466001
8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 522083/87
1000 Berlin 30, Keithstraße 2–4, Tel. (030) 2490 86
6800 Mannheim, Roßlauer Weg 2–4, Tel. (0621) 70050
7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 70241
7517 Waldbronn 2, Hewlett-Packard Straße, Tel. (07243) 602-1

Hewlett-Packard (Schweiz) AG:
Allmend 2, CH-8967 Widen, Tel. (057) 31 21 11

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:
Wagramerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 23 65 11

Hewlett-Packard S.A., Europa-Zentrale:
7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz

SERVICENIEDERLASSUNGEN:

Hewlett-Packard GmbH:
6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1

Hewlett-Packard (Schweiz) AG:
Allmend 2, CH-8967 Widen, Tel. (057) 31 21 11

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:
Wagramerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 23 65 11

HEWLETT PACKARD

00075-90006 German - 10.82. - CP

Printed in Singapore 10/82