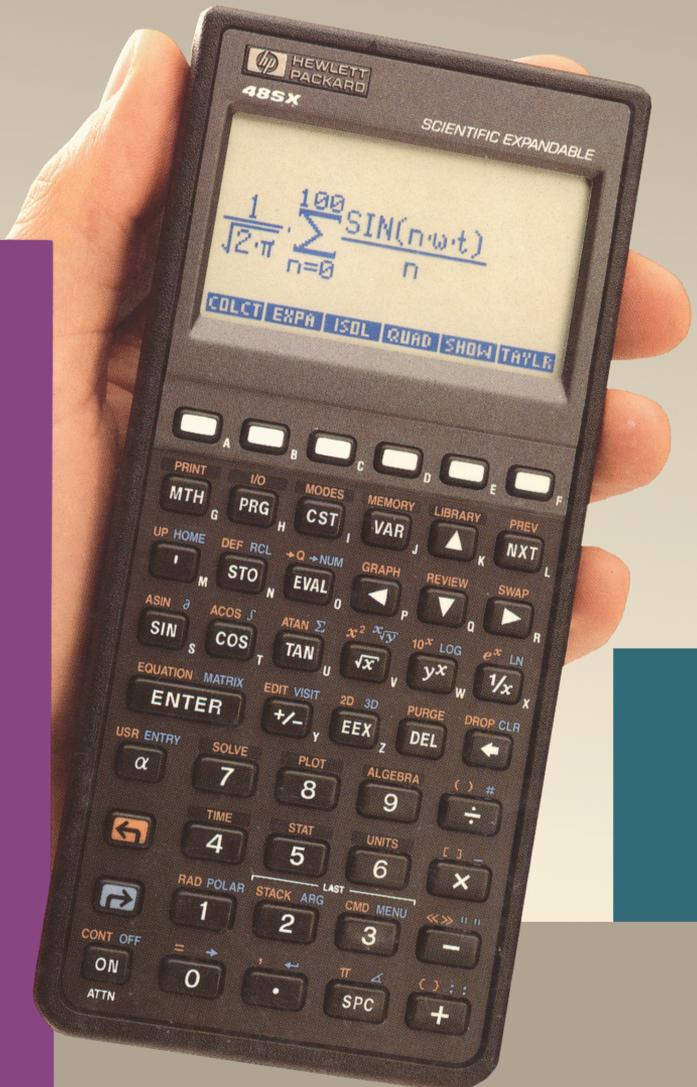


HP 48SX

Erweiterbarer wissenschaftlicher
Taschenrechner



**Benutzerhandbuch
Band II**

Erweiterbarer wissenschaftlicher Taschenrechner HP 48SX

**Benutzerhandbuch
Band II**



Ausgabe 1 April 1990
Bestell-Nr. 00048-90007

Zur Beachtung Informationen

Garantiebestimmungen finden Sie auf Seite 731 und 733.

Dieses Handbuch und die darin enthaltenen Beispiele werden "so wie sie sind" zur Verfügung gestellt und können ohne vorherige Ankündigung geändert werden. Hewlett-Packard haftet nicht für irgendwelche Fehler und mittelbare oder unmittelbare Schäden in Zusammenhang mit der Lieferung, Funktionsfähigkeit oder Verwendung dieses Handbuchs oder der darin enthaltenen Beispiele.

• Hewlett-Packard Co. 1990. Alle Rechte vorbehalten. Bearbeitung oder Übersetzung dieses Handbuchs ist nur nach vorheriger schriftlicher Genehmigung durch Hewlett-Packard gestattet, soweit nicht durch das Urheberrecht erlaubt.

Die Programme, die Ihren Taschenrechner steuern, sind urheberrechtlich geschützt und alle Rechte sind vorbehalten. Vervielfältigung, Bearbeitung oder Übersetzung dieser Programme ist ebenfalls nur nach vorheriger schriftlicher Genehmigung durch Hewlett-Packard gestattet.

• Trustees of Columbia University in the City of New York, 1989. Es wird jeder Einzelperson oder Institution die Erlaubnis erteilt, Kermit-Software zu verwenden, zu kopieren oder weiterzuveräußern, so lange dies nicht der Erzielung eines Gewinns dient und sofern dieser urheberrechtliche Hinweis bewahrt bleibt.

Drucklegende

1. Ausgabe

April 1990

Teile-Nr. 00048-90008

Inhaltsverzeichnis

Teil 4: Programmierung

- 25** **504** **Grundlagen der Programmierung**
- 506** Eingabe und Ausführung eines Programms
- 506** Ein Programm eingeben
- 508** Ausführung eines Programmes
- 509** Ein Programm editieren
- 510** Verwendung von lokalen Variablen
- 516** Programme, die mit Daten im Stack arbeiten
- 518** Verwendung von Unterprogrammen
- 521** Einzelschrittausführung eines Programms
- 522** Einzelschrittausführung beginnend am Anfang des
 Programms
- 524** Einzelschrittausführung einer beliebigen Stelle im
 Programm
- 524** Einzelschrittausführung von Unterprogrammen

- 26** **526** **Tests und Bedingte Strukturen**
- 528** Programmtests
- 529** Vergleichsfunktionen
- 531** Logische Funktionen
- 532** Objekttypen testen
- 532** Bedingte Strukturen
- 532** Die Struktur IF...THEN...END
- 535** Die Struktur IF...THEN...ELSE...END
- 536** Die Struktur CASE...END
- 538** Bedingte Befehle
- 539** Der Befehl IFT (Wenn-dann-Ende)

	539	Die Funktion IFTE
27	540	Aufbau von Programmschleifen
	540	Aufbau von bestimmten Programmschleifen
	541	Die Struktur START...NEXT
	543	Die Struktur START...STEP
	545	Die Struktur FOR...NEXT
	547	Die Struktur FOR...STEP
	549	Aufbau von unbestimmten Programmschleifen
	549	Die Struktur DO...UNTIL...END
	551	Die Struktur WHILE...REPEAT...END
	553	Schleifenzähler (INCR und DECR)
28	555	Flags
	555	Flag-Typen
	556	Setzen, Rücksetzen und Testen von Flags
	558	Aufrufen und Speichern der Flag-Zustände
	559	Aufrufen der Flag-Zustände
	559	Speichern der Flag-Zustände
29	560	Dialogprogramme
	561	Unterbrechung des Programmablaufs für eine Dateneingabe
	562	Der Befehl PROMPT
	564	Der Befehl BEEP
	565	Die Befehle DISP, HALT und FREEZE
	566	Der Befehl INPUT
	575	Kennzeichnen der ausgegebenen Daten
	575	Verwenden von markierten Objekten als ausgegebene Daten
	576	Verwenden von Zeichenketten-Befehlen zur Kennzeichnung ausgegebener Daten
	578	Unterbrechen eines Programms, um Daten anzuzeigen
	578	Verwenden von Menüs in Programmen
	578	Anzeige eines internen Menüs
	579	Benutzerspezifische Menüs in Programmen
	583	Aufbau eines temporären Menüs

	584	Befehle, die eine Tastenadresse liefern
	584	Der Befehl WAIT mit dem Argument 0
	584	Der Befehl WAIT mit dem Argument - 1
	584	Der Befehl KEY
	585	Abschalten des HP48 aus einem Programm
30	586	Fehlerbehandlung
	588	Die Struktur IFERR...THEN...END
	590	Die Struktur IFERR...THEN...ELSE...END
	592	Benutzerdefinierte Fehler
31	593	Weitere Programmierbeispiele
	594	Fibonacci-Zahlen
	594	FIB1 (Fibonacci-Zahlen, Rekursive Version)
	596	FIB2 (Fibonacci-Zahlen, Schleifenversion)
	597	FIBT (Vergleich der Programmausführungszeit)
	600	Anzeigen einer Binärzahl
	600	PAD (Auffüllen mit führenden Leerstellen)
	602	PRESERVE (Sichern und Wiederherstellen des vorherigen Zustands)
	603	BDISP (Binäranzeige)
	607	Medianwert von statistischen Daten
	608	SORT (Ordnen einer Liste)
	610	LMED (Medianwert einer Liste)
	612	MEDIAN (Medianwert von statistischen Daten)
	616	Vollständiges Erweitern und Zusammenfassen
	616	MULTI (Mehrfache Ausführung)
	618	EXCO (Vollständiges Erweitern und Zusammenfassen)
	620	Bestimmung des größten oder kleinsten Elementes in einem Feld
	620	MNX (Bestimmung des kleinsten oder des größten Elementes in einem Feld — Verfahren 1)
	624	MNX2 (Bestimmung des größten oder des kleinsten Elementes in einem Feld — Verfahren 2)
	627	Prüfung von Programmargumenten
	628	NAMES (Enthält die Liste genau zwei Namen?)
	630	VFY (Überprüfung von Argumenten für Programme)
	634	Bessel-Funktionen

- 637 Bewegte Darstellung einer Folge von Taylor-Polynomen
- 637 Erzeugen einer Sinuskurve und Umwandlung in ein Grafikobjekt
- 638 Überlagerung einer Folge von Taylor-Polynomen
- 640 Bewegte Darstellung von Taylor-Polynomen
- 642 Anwendung von Statistikdaten und grafischen Darstellungen in Programmen
- 647 Bewegte Darstellung einer Grafik

Teil 5: Drucken, Datenübertragung und Einsteckmodule

- 32**
 - 652 **Drucken**
 - 652 Drucken mit einem Drucker HP 82240B
 - 654 Druckformate
 - 655 Grunddruckbefehle
 - 657 Druck einer Textzeichenfolge
 - 657 Druck von grafischen Objekten.
 - 657 Zweizeiliger Druck
 - 657 Einstellen der Verzögerungszeit
 - 658 Der Zeichensatz des HP48
 - 658 Senden von Escape-Sequenzen und Steuercodes
 - 659 Speichern von Daten im Pufferspeicher des Druckers
 - 660 Drucken mit einem Infrarotdrucker HP 82240A
 - 660 Drucken über den seriellen Anschluß
 - 662 Die Variable PRTPAR

- 33**
 - 663 **Datenübertragung vom und zum HP48**
 - 664 Arten von Daten, die Sie übertragen können
 - 665 Das E/A-Menü zur Ein- und Ausgabe von Daten
 - 668 Lokaler Modus und Servermodus
 - 669 Einstellung der E/A-Parameter
 - 669 Das Menü SETUP
 - 671 Die Variable IOPAR
 - 672 Übertragung von Daten zwischen zwei HP48
 - 674 Übertragung von Daten zwischen einem Computer und dem HP 48

- 674** Kabelverbindung
- 675** Übertragung von Daten
- 677** Sichern aller Daten im Speicher des HP48
- 679** Umsetzen von Zeichen (TRANSIO)
- 681** Mehr zu den Dateinamen
- 682** Fehler
- 682** Übertragung im ASCII-Modus und im Binär-Modus
- 684** Senden von Befehlen an einen Server (PKT)
- 685** Befehle für die serielle Ein- und Ausgabe

34

- 688** **Verwendung von Einsteckkarten und Bibliotheken**
- 688** Speicherarten
- 689** Einbau und Ausbau von Einsteckkarten
- 693** RAM-Karten
- 693** Vorbereiten der Karte für den Einbau
- 697** Anwendungen für RAM-Karten
- 698** Verwendung von RAM-Karten zur Erweiterung des Benutzerspeichers (Eingebundener Speicher)
- 699** Verwendung von RAM-Karten für die Sicherung von Daten (Unabhängiger Speicher)
- 699** Sichern von Objekten in einen unabhängigen Speicher
- 701** Zugreifen auf Sicherungsobjekte
- 702** Sichern von Daten in den Benutzerspeicher (Port 0)
- 703** Sichern des gesamten Speichers
- 704** Freigeben des eingebundenen Speichers
- 706** Verwenden von ROM-Steckkarten und Bibliotheken
- 707** Anbinden einer Bibliothek an ein Inhaltsverzeichnis
- 707** Zugreifen auf Bibliotheksoperationen (Das Menü LIBRARY)
- 709** Weitere Befehle für den Zugriff auf Bibliotheken

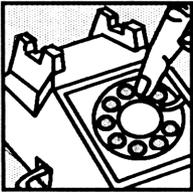
Teil 6: Anhänge und Verzeichnisse

- A**
- 712** **Unterstützung, Batterien und Wartung**
 - 712** Unterstützung bei der Arbeit mit dem Taschenrechner
 - 712** Antworten auf häufige Fragen
 - 716** Umgebungsbedingungen
 - 717** Was beim Austausch der Batterien zu beachten ist
 - 718** Batterien wechseln
 - 718** Batterietypen
 - 718** Batterien des Taschenrechners wechseln
 - 720** Batterie der RAM-Karte wechseln
 - 722** Überprüfung der Taschenrechnerfunktion
 - 724** Selbsttest
 - 725** Testen der Tastatur
 - 726** Testen des Port-RAM
 - 728** Schleifentest des IR-Anschlusses
 - 729** Schleifentest des seriellen Anschlusses
 - 731** Begrenzte Ein-Jahres-Garantie
 - 732** Wenn der Taschenrechner repariert oder gewartet werden muß
 - 733** Störsicherheit
- B**
- 735** **Meldungen**
- C**
- 752** **Zeichencodes des HP 48**
- D**
- 755** **Menünummern**
- E**
- 757** **Zusammenstellung der System-Flags des HP 48**
 - 766** **Verzeichnis der Operationen**
 - 881** **Index**

Teil 4

Programmierung

Grundlagen der Programmierung



Unter einem Programm versteht man ein Objekt, das durch die Begrenzungszeichen `※ ※` definiert ist. Ein Programm besteht selbst aus aus Objekten und Befehlen, deren Ausführung bis zur Ausführung des Programms *verschoben* wird. Da ein Programm ein Objekt ist, gilt:

- Es kann im Stack abgelegt werden.
- Es kann in einer Variablen gespeichert werden.
- Es kann wiederholt ausgeführt werden.
- Es kann durch ein anderes Programm ausgeführt werden.

Im folgenden Beispiel wird das Volumen einer Kugel bestimmt, wobei die Berechnung zuerst über die Tastatur und anschließend mit einem Programm ausgeführt wird.

Beispiel: Berechnungen über die Tastatur und mit einem Programm. Das Volumen einer Kugel mit dem Radius r wird mit folgender Gleichung berechnet:

$$V = \frac{4}{3} \pi r^3$$

Wenn Sie die Berechnung einmal ausführen wollen, können Sie das mit folgenden Tastendrücken tun. (Vorausgesetzt ist dabei, daß der Radius bereits im Stack steht.)

3 y^x \leftarrow π \times 4
 \times 3 \div \rightarrow \rightarrow NUM

Jedes Mal, wenn Sie eine Befehlstaste drücken, wird der Befehl unverzüglich ausgeführt; das Zwischenergebnis wird im Stack abgelegt.

Wenn Sie die Volumina von mehreren Kugeln berechnen wollen, können Sie ein Programm schreiben. Das folgende Programm setzt voraus, daß der Radius zu Beginn der Programmausführung im Stack steht:

« 3 ^ π * 4 * 3 / \rightarrow NUM »

Nach Eingabe der Begrenzungszeichen « » (durch Drücken von \leftarrow « ») verwenden Sie dieselben Tastendrücke wie vorher, um die folgenden Objekte und Befehle einzugeben. Die Objekte und Befehle werden allerdings nur einfach in die Befehlszeile geschrieben — ihre Ausführung findet erst statt, wenn Sie das Programm selbst ablaufen lassen.

Da es sich bei dem Programm um ein Objekt handelt, können Sie es im Stack ablegen und in einer Variablen speichern. Um das Programm im Stack abzulegen, müssen Sie **ENTER** drücken. Wenn Sie das Programm in einer Variablen mit dem Namen *VOL* speichern wollen, können Sie das durch Eingabe von \square VOL **STO** tun. Jetzt können Sie das Volumen einer beliebigen Kugel berechnen, indem Sie einfach den Radius in den Stack eingeben und *VOL* ablaufen lassen. (Wählen sie das Menü VAR und drücken Sie **VOL**). Sie können *VOL* so oft ablaufen lassen, wie Sie wollen; es verhält sich wie ein eingebauter Befehl.

VOL ist ein Programm der einfachsten Form; es besteht aus einer Folge von Objekten und Befehlen in der gleichen Reihenfolge, in der Sie sie über die Tastatur eingeben würden. In den nächsten Kapiteln werden Sie weiterentwickelte Merkmale des HP 48 kennenlernen, die Sie beim Programmieren nutzen können.

- Ausdrücke mit logischen Bedingungen (Kapitel 26).
- Schleifenstrukturen (Kapitel 27).
- Flags (Kapitel 28).

- Interaktive Programme (Kapitel 29).
- Fehlerbehandlung (Kapitel 30).

In diesem Kapitel werden die Grundzüge der Programmierung des HP 48 behandelt:

- Eingabe und Ausführung von Programmen.
- Editieren von Programmen.
- Verwendung von lokalen Variablen in Programmen.
- Manipulation von Daten im Stack in Programmen.
- Verwendung von Unterprogrammen.
- Einzelschrittausführung von Programmen.

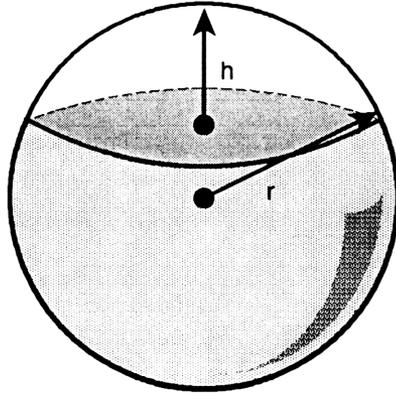
Das Handbuch *Programmer's Reference Manual* für den HP 48 (Bestellnummer 00048-90054) enthält hilfreiche Informationen zur Programmierung Ihres Taschenrechners einschließlich der vollständigen Syntax für alle Befehle des HP 48.

Eingabe und Ausführung eines Programms

Ein Programm eingeben

Drücken Sie $\boxed{\leftarrow}$ $\boxed{\ll \gg}$, um den Anfang eines Programms zu kennzeichnen. Es erscheint die Meldung `PRG`, die den Programmeingabemodus anzeigt. In diesem Modus wird durch die Betätigung einer Befehlstaste der Name dieses Befehles in die Befehlszeile geschrieben. (Sie können den Befehlsnamen auch über die Buchstabentastatur in die Befehlszeile schreiben.) Nur nichtprogrammierbare Operationen wie $\boxed{\leftarrow}$ und $\boxed{\text{VAR}}$ werden ausgeführt.

Mit dem folgenden Programm *SPH* können Sie das Volumen einer Kugelkalotte mit dem Radius r und der Höhe h berechnen.



Das Volumen ergibt sich aus der Beziehung $V = \frac{1}{3} \pi h^2 (3r - h)$.

In diesem und den nächsten Kapiteln über die Programmierung werden “Stackdiagramme” verwendet, um zu zeigen, welche Objekte vor einer Programmausführung im Stack stehen müssen und welche Ergebnisse vom Programm in den Stack übertragen werden. Es folgt das Stackdiagramm für *SPH*.

Argumente	Ergebnisse
2: 1:	2: 1: <i>Volumen</i>

Aus der Übersicht ist zu erkennen, daß *SPH* keine Argumente vom Stack nimmt und das Volumen der Kugelkalotte nach Ebene 1 überträgt. (*SPH* setzt voraus, daß Sie den Zahlenwert des Radius in der Variablen *R* und den Zahlenwert für die Höhe in der Variablen *H* gespeichert haben.)

Die ausgedruckten Programme enthalten in der linken Spalte die einzelnen Programmschritte und in der rechten Spalte die dazu gehörenden Erläuterungen. Beachten Sie bitte, daß Sie zur Eingabe des Programms *entweder* die Befehlstasten betätigen *oder* die Befehlsnamen eingeben können. In diesem ersten ausgedruckten Programm sind auch die einzelnen Tasten dargestellt, die zu betätigen sind.

Programm:	Tasten:	Erläuterungen:
⌘	\leftarrow \ll \gg	Kennzeichnet den Anfang des Programms.
' 1/3	\square 1 \div 3	Erster Rechenschritt in dem algebraischen Ausdruck, der das Volumen bestimmt.
* π *H ²	\times \leftarrow π \times H y^x 2	Multipliziert mit πh^2 .
* (3*R-H) '	\times \leftarrow () 3 \times R $-$ H \rightarrow \rightarrow	Multipliziert mit $3r - h$, schließt damit die Berechnung ab und beendet den Ausdruck.
\rightarrow NUM	\rightarrow \rightarrow NUM	Wandelt π in einen Zahlenwert um.
⌘		Beendet das Programm.
	ENTER	Legt das Programm im Stack ab.
	\square SPH STO	Speichert das Programm in der Variablen SPH.

Ausführung eines Programmes

Es gibt mehrere Möglichkeiten, SPH ablaufen zu lassen:

- Schreiben Sie SPH in die Befehlszeile; drücken Sie anschließend **ENTER**.
- Wählen Sie das Menü VAR; drücken Sie danach \square SPH \square .
- Befindet sich das Programm oder der Programmname bereits in Ebene 1, betätigen Sie **EVAL**.

Beispiel: Ausführung eines Programms aus dem Menü VAR.

Verwenden Sie SPH, um das Volumen einer Kugelkalotte mit dem Radius $r = 10$ mm und der Höhe $h = 3$ mm zu berechnen.

Speichern Sie im ersten Schritt die Daten in den entsprechenden Variablen. Wählen sie dann das Menü VAR und lassen Sie das Programm laufen. Das Ergebnis in Ebene 1 des Stacks abgelegt.

10 \square R **STO**
 3 \square H **STO**
VAR \square SPH \square

1: 254,469004942
 H R SPH

Ein Programm editieren

Wenn Sie ein Programm editieren, müssen Sie dieselben Regeln wie bei der Editierung eines anderen Objektes beachten (Einzelheiten hierzu finden Sie im Abschnitt “Anzeigen und Bearbeiten von Objekten” auf Seite 70).

Beispiel: Ein Programm editieren. Ändern Sie *SPH* so, daß es die Zahl in Ebene 1 in der Variablen *H* speichert und die Zahl in Ebene 2 in der Variablen *R* ablegt.

Verwenden Sie das Menü VAR und [VISIT], um *SPH* zur Editierung in die Befehlszeile zu holen.

[VAR]
[] SPH
[] [VISIT]

```
◀ '1/3*π*H^2*(3*R-H  
)' →NUM  
▶  
◀SKIP ▶SKIP ▶DEL DEL▶ INS ◀↑STK
```

Bewegen Sie den Cursor hinter das erste Begrenzungszeichen des Programms, und fügen Sie die neuen Programmschritte ein.

[▶] [] H [▶] [STO]
[] R [▶] [STO]

```
◀ 'H' STO 'R' STO ◀1/3...  
)' →NUM  
▶  
◀SKIP ▶SKIP ▶DEL DEL▶ INS ◀↑STK
```

Speichern Sie die editierte Version von *SPH* in der Variablen.

Kontrollieren Sie, ob die Änderungen gesichert worden sind, indem Sie *SPH* in die Befehlszeile zurückrufen.

[ENTER]
[] SPH
[] [VISIT]

```
◀ 'H' STO 'R' STO '  
1/3*π*H^2*(3*R-H)'  
→NUM  
▶  
◀SKIP ▶SKIP ▶DEL DEL▶ INS ◀↑STK
```

Weitere Änderungen sind nicht erforderlich; Sie können mit [ATTN] die Editierung abbrechen oder das Programm mit [ENTER] nochmals sichern.

Die geänderte Version von *SPH* holt jetzt zwei Argumente aus dem Stack: die Höhe von Ebene 1 und den Radius von Ebene 2.

Verwendung von lokalen Variablen

Das Programm *SPH* im letzten Abschnitt verwendet globale Variablen, um Daten zu speichern und aufzurufen. Es gibt einige Nachteile bei der Verwendung von globalen Variablen in Programmen:

- Nach der Programmausführung müssen Sie die globalen Variablen löschen, die Sie nicht mehr benötigen, wenn Sie das Menü VAR aufheben und Benutzerspeicher freimachen wollen.
- Sie müssen Daten vor der Programmausführung ausdrücklich in globalen Variablen speichern oder Sie müssen dies mit Hilfe des Befehls STO vom Programm erledigen lassen.

In diesem Abschnitt werden Sie sehen, wie man mit *lokalen Variablen* die Nachteile von globalen Variablen in Programmen vermeiden kann. Lokale Variablen sind vorübergehend verwendete Variable, die von einem Programm erzeugt werden. Sie existieren nur solange, wie das Programm ausgeführt wird und können außerhalb des Programms nicht verwendet werden. Sie erscheinen auf keinen Fall im Menü VAR.

Wenn Sie lokale Variablen erzeugen wollen, müssen Sie die folgende Sequenz von Befehlen und Objekten verwenden; diese Sequenz wird als *Struktur lokaler Variable* bezeichnet:

1. Den Befehl \rightarrow (drücken sie  ).
2. Einen oder mehrere Variablennamen.
3. Eine Prozedur (einen algebraischen Ausdruck oder ein Programm), der die Namen enthält. Diese Prozedur wird als *definierende Prozedur* bezeichnet.

Die Struktur hat folgende Form:

$$\llcorner \rightarrow \text{Name}_1 \text{Name}_2 \dots \text{Name}_n \llcorner \text{Programm} \gg$$

oder

$$\llcorner \rightarrow \text{Name}_1 \text{Name}_2 \dots \text{Name}_n \text{ 'Algebraischer Ausdruck' } \gg$$

Bei Ausführung des Befehls \rightarrow in einem Programm werden n Werte aus dem Stack geholt und den Variablen $Name_1, Name_2, \dots, Name_n$ zugewiesen. Enthält der Stack zum Beispiel folgende Daten:

{ HOME }	
4:	
3:	10
2:	6
1:	20
PDATE PRDE WWP MWTR WECTR BMSB	

dann:

- erzeugt \rightarrow \exists die lokale Variable $a = 20$.
- erzeugt \rightarrow \exists b die lokalen Variablen $a = 6$ und $b = 20$.
- erzeugt \rightarrow \exists b c die lokalen Variablen $a = 10$, $b = 6$ und $c = 20$.

Die definierende Prozedur verwendet dann die lokalen Variablen zur Ausführung der Berechnungen.

(In diesem Handbuch erscheinen die Namen der lokalen Variablen in Kleinbuchstaben.)

Das folgende Programm *SPHLV* verwendet lokale Variablen, um das Volumen einer Kugelkalotte zu berechnen. Die definierende Prozedur ist ein algebraischer Ausdruck.

Argumente	Ergebnis
2: r 1: h	2: 1: <i>Volumen</i>

Programm:**Kommentar:**

«

→ r h

Erzeugt die lokalen Variablen r und h für den Radius der Kugel und die Höhe der Kalotte.

'1/3*π*h^2*(3*r-h)'

Definierende Prozedur: in diesem Programm ist die definierende Prozedur für die Struktur lokaler Variablen ein algebraischer Ausdruck.

→NUM

Wandelt π in eine Zahl um.

»

[ENTER] **[]** SPHLV **[STO]**Speichert das Programm in der Variablen *SPHLV*.

Beispiel: Ausführen eines Programms, das lokale Variablen verwendet. Verwenden Sie *SPHLV*, um das Volumen einer Kugelkalotte mit dem Radius $r = 10$ mm und der Höhe $h = 3$ mm zu berechnen.

Geben Sie die Daten in der richtigen Reihenfolge in den Stack ein, wählen Sie dann das Menü VAR und starten Sie das Programm.

10 **[ENTER]** 3
[VAR] SPHLV

1:	254,469004942
SPHLV	H
R	SPH

Dieses Beispiel zeigt die Vorteile einer Struktur mit lokalen Variablen:

- Der Befehl → speichert den (die) Wert(e) vom Stack in der (den) entsprechenden Variablen — Sie brauchen nicht ausdrücklich STO auszuführen.
- Lokale Variable verschwinden automatisch, wenn die definierende Prozedur abgeschlossen ist, für die sie erzeugt worden sind. Daraus folgt, daß lokale Variablen nicht im Menü VAR erscheinen und den Anwenderspeicher nur beim Ablauf des Programms belegen.
- Lokale Variablen existieren nur innerhalb ihrer definierenden Prozedur — unterschiedliche Strukturen lokaler Variablen können dieselben Variablennamen verwenden, ohne daß es zu Konflikten kommt.

Auswertung von lokalen Namen. Lokale Namen werden anders als globale Namen ausgewertet. Bei der Auswertung eines globalen Namens wird das in der entsprechenden Variablen gespeicherte Objekt selbst ausgewertet. (Sie haben gesehen, wie in globalen Variablen gespeicherte Programme automatisch ausgewertet werden, wenn der Name ausgewertet wird.)

Bei der Auswertung eines lokalen Namens wird das in der entsprechenden Variablen gespeicherte Objekt im Stack abgelegt, aber *nicht* ausgewertet. Enthält eine lokale Variable eine Zahl, hat das dieselbe Wirkung wie die Auswertung eines globalen Namens, da das Ablegen einer Zahl im Stack gleichbedeutend mit seiner Auswertung ist.

Wenn eine lokale Variable allerdings ein Programm, einen algebraischen Ausdruck oder den Namen einer globalen Variablen enthält, *muß* das Objekt *ausdrücklich* (durch Ausführen von EVAL) *ausgewertet werden*, nachdem es im Stack abgelegt worden ist.

Gültigkeitsbereich von lokalen Variablen. Lokale Variablen existieren *nur* in der Prozedur für die sie definiert sind. Das folgende Musterprogramm zeigt die Verfügbarkeit von lokalen Variablen in *verschachtelten* definierenden Prozeduren (Prozeduren innerhalb von Prozeduren).

Programm:	Kommentar:
«	Beginnt das äußere Programm.
.	Für diese beliebigen Programmschritte stehen keine lokalen Variablen zur Verfügung.
→ a b c	Erzeugt die lokalen Variablen <i>a</i> , <i>b</i> und <i>c</i> .
«	Beginnt die definierende Prozedur (ein Programm) für die lokalen Variablen <i>a</i> , <i>b</i> und <i>c</i> . Diese Prozedur läuft innerhalb des äußeren Programms. In dieser Prozedur sind die lokalen Variablen <i>a</i> , <i>b</i> und <i>c</i> verfügbar.

a b + c +

→ d e f

Definiert die lokalen Variablen d , e und f .

'

Beginn der definierenden Prozedur (ein algebraischer Ausdruck) für die lokalen Variablen d , e und f . Diese Prozedur läuft innerhalb der definierenden Prozedur für die lokalen Variablen a , b und c . Damit sind in dieser Prozedur die lokalen Variablen a , b , c , d , e und f verfügbar.

a/(d*e+f)

'

Ende der definierenden Prozedur für die lokalen Variablen d , e , f . Die lokalen Variablen d , e und f existieren nicht mehr.

a c / -

Die lokalen Variablen a , b und c bleiben verfügbar.

⌘

Ende der definierenden Prozedur für die lokalen Variablen a , b und c . Die lokalen Variablen a , b und c existieren nicht mehr.

.

Für diese beliebigen Programmschritte sind keine lokalen Variablen verfügbar.

⌘

Beendet das äußere Programm.

Da die lokalen Variablen a , b und c bereits existieren, wenn die definierende Prozedur für die lokalen Variablen d , e und f ausgeführt wird, sind sie für die Verwendung auch in dieser Prozedur verfügbar.

Nehmen Sie nun einmal an, daß die definierende Prozedur für die lokalen Variablen d , e und f ein Programm aufruft, daß Sie vorher geschrieben und in der globalen Variablen PI gespeichert haben.

Programm:

Kommentar:

```

«
. . . . .
→ a b c
«
  a b + c +
→ d e f
'
P1+a/(d*e+f)
'
. . . . .
»
. . . . .
»

```

Definiert die lokalen Variablen *d*, *e* und *f*.

Beginnt die definierende Prozedur für die lokalen Variablen *d*, *e* und *f*.

Die definierende Prozedur ruft das in *P1* gespeicherte Programm auf.

Ende der definierenden Prozedur für die lokalen Variablen *d*, *e* und *f*.

Die sechs lokalen Variablen sind *nicht* im Programm *P1* verfügbar, weil sie beim Schreiben des Programms *P1* nicht existierten. Die in den lokalen Variablen gespeicherten Objekte stehen dem Programm *P1* nur dann zur Verfügung, wenn Sie diese Objekte als Argumente für *P1* im Stack abgelegt oder diese Objekte in globalen Variablen gespeichert haben.

Umgekehrt zum Beispiel kann das Programm *P1* eine eigene Struktur lokaler Variablen mit den lokalen Variablen *a*, *c* und *f* anlegen, ohne daß ein Konflikt mit den lokalen Variablen desselben Namens in der Prozedur entsteht, die *P1* aufruft.

Programme, die wie benutzerdefinierte Funktionen wirken. In diesem Kapitel ist beschrieben, daß die definierende Prozedur einer Struktur lokaler Variablen entweder ein algebraischer Ausdruck oder ein Programm sein kann. In Kapitel 10 haben Sie gelernt, daß eine benutzerdefinierte Funktion ein Programm ist, das ausschließlich aus einer Struktur lokaler Variablen besteht, deren definierende Prozedur ein algebraischer Ausdruck ist.

Ein Programm, das mit einer Struktur lokaler Variablen beginnt, deren definierende Prozedur ein Programm ist, wirkt auf zwei Arten wie eine benutzerdefinierte Funktion: Es verarbeitet numerische oder symbolische Argumente, und es nimmt diese Argumente entweder aus dem Stack oder verwendet algebraische Syntax. Das Programm besitzt allerdings *keine* Ableitung. (Wie algebraische Definitionsverfahren darf das Definitionsprogramm *nur ein* Ergebnis in den Stack zurück übertragen.)

Der Vorteil bei der Verwendung eines Programms als definierende Prozedur einer Struktur lokaler Variablen liegt darin, daß ein Programm Befehle enthalten kann, die in algebraischen Ausdrücken nicht zulässig sind. Zum Beispiel sind die in Kapitel 27 beschriebenen Schleifenstrukturen in algebraischen Ausdrücken nicht zulässig. Das Programm *BER* in Kapitel 31 berechnet die Näherung einer Besselfunktion auf 12 Stellen genau. *BER* verwendet eine Struktur lokaler Variablen, deren definierende Struktur ein RPN-Programm ist. Dieses Programm enthält eine FOR...STEP- Struktur und eine verschachtelte IF...THEN...ELSE...END- Struktur. *BER* ist nicht differenzierbar, das Beispiel in Kapitel 31 zeigt aber, daß es seine Argumente entweder aus dem Stack nehmen oder algebraische Syntax verwenden kann.

Programme, die mit Daten im Stack arbeiten

Die Programme *SPH* (Seite 508) und *SPHLV* (Seite 512) in diesem Kapitel verwenden Variable für das Speichern und Aufrufen von Daten. Ein alternatives Programmierverfahren arbeitet stattdessen mit Zahlen, die im Stack stehen. Mit diesem Verfahren erzeugte Programme sind im allgemeinen schneller. Das Verfahren der Stackmanipulation hat jedoch auch einige Nachteile:

- Beim Schreiben eines Programms muß die Position der Daten im Stack verfolgt werden. Zum Beispiel müssen Datenargumente dupliziert werden, wenn sie von mehr als einem Befehl verwendet werden.

- Ein Programm, das mit Daten im Stack arbeitet, ist im allgemeinen nicht so einfach zu lesen und schwerer zu verstehen als ein Programm, das mit Variablen arbeitet.

Das folgende Programm *SPHSTACK* verwendet das Verfahren der Stackmanipulation, um das Volumen einer Kugelkalotte zu berechnen. (Es führt dieselbe Berechnung aus wie *SPH* und *SPHLV*.)

Argumente	Ergebnisse
2: r 1: h	2: 1: <i>Volumen</i>

Programm:

Kommentar:

«

DUP

Legt eine Kopie der Zahl in Ebene 1 (die Höhe) in Ebene 2 ab

ROT

Rotiert die Zahl, die sich jetzt in Ebene 3 befindet (den Radius) nach Ebene 1.

3 *

Multipliziert den Radius mit 3.

SWAP -

Bringt durch Vertauschen die Höhe nach Ebene 1 und führt eine Subtraktion zur Berechnung von $3r - h$ aus.

SWAP SQ *

Bringt durch Vertauschen die Kopie der Höhe nach Ebene 1, quadriert sie und multipliziert sie mit $3r - h$.

π * 3 /

Multipliziert mit π und teilt durch 3. Damit ist die Berechnung beendet.

→NUM

Wandelt π in eine Zahl um.

»

ENTER **SPHSTACK** **STO**

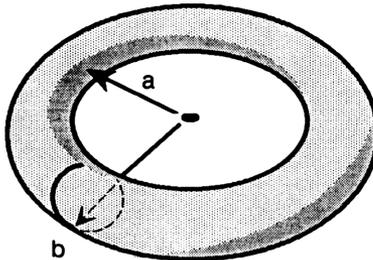
Legt das Programm im Stack ab und speichert es anschließend in *SPHSTACK*.

Verwendung von Unterprogrammen

Sie erinnern sich sicher daran, daß ein Programm aus Objekten und Befehlen besteht, die beim Ablauf des Programms ausgeführt werden. Da ein Programm selbst ein Objekt darstellt, kann es wiederum von einem anderen Programm verwendet werden. Wird das Programm *B* vom Programm *A* verwendet, ruft das Programm *A* das Programm *B* auf, und das Programm *B* bildet ein *Unterprogramm* im Programm *A*.

In diesem Abschnitt werden zwei Programme vorgestellt, die die Funktion von Unterprogrammen veranschaulichen.

Mit dem ersten Programm *TORSA* läßt sich die Oberfläche eines Torus mit dem inneren Radius *a* und dem äußeren Radius *b* berechnen. *TORSA* wird im zweiten Programm als Unterprogramm verwendet.



Die Oberfläche wird mit folgender Beziehung berechnet:

$$A = \pi^2 (b^2 - a^2)$$

Es folgt das Stackdiagramm und die Programmliste für *TORSA*.

Argumente	Ergebnisse
2: a 1: b	2: 1: Fläche

Programm:

Kommentar:

⌘

→ a b

Erzeugt die lokalen Variablen *a* und *b*.

'π^2*(b^2-a^2)'

Definierende Prozedur der Struktur lokaler Variablen.

→NUM

Wandelt π in eine Zahl um.

⌘

ENTER

Legt das Programm im Stack ab.

TORSA **STO**

Speichert das Programm in *TORSA*.

Das Programm *TORSV* berechnet das Volumen eines Torus. Es ruft *TORSA* auf, das einen Teil der Berechnung auszuführen.

Die Formel für das Volumen des Torus lautet:

$$V = \frac{1}{4} \pi^2 (a + b)(b - a)^2$$

Sie läßt sich auch in dieser Form schreiben:

$$V = \frac{1}{4} \pi^2 (b^2 - a^2)(b - a)$$

Die Größe $\pi^2 (b^2 - a^2)$ in dieser Gleichung ist die Oberfläche eines Torus und kann durch Ausführung von *TORSA* berechnet werden.

Es folgt das Stackdiagramm für *TORSV*.

Argumente	Ergebnisse
2: <i>a</i> 1: <i>b</i>	2: 1: <i>Volumen</i>

Programm:

Kommentar:

«

→ *a b*

Erzeugt die lokalen Variablen *a* und *b*.

«

Beginnt die definierende Prozedur der Struktur der lokalen Variablen (ein Programm).

a b TORS A

Legt die in *a* und *b* gespeicherten Zahlen als Argumente für *TORS A* im Stapel ab und ruft danach *TORS A* zur Berechnung der Fläche $\pi^2(b^2 - a^2)$ auf.

*b a - * 4 /*

Dieser Rechenschritt schließt die Berechnung ab.

»

Ende der definierenden Prozedur.

»

Beendet das Programm.

ENTER

Legt das Programm im Stack ab.

□ TORSV **STO**

Speichert das Programm in *TORSV*.

TORSV ruft das Programm *TORS A* auf, um einen Teil der Volumeberechnung auszuführen. *TORS A* ist ein Unterprogramm in *TORSV*. Umgekehrt kann ein anderes Programm auch *TORSV* aufrufen.

Beispiel: Ausführung eines Programms, das ein Unterprogramm verwendet. Verwenden Sie *TORSV*, um das Volumen eines Torus mit dem inneren Radius $a = 6$ cm und dem äußeren Radius $b = 8$ cm zu berechnen.

Legen Sie die Daten entsprechend dem Stackdiagramm in den Stack. Wählen Sie das Menü VAR und lassen Sie das Programm ablaufen.

6 [ENTER] 8
 [VAR] TORSV

1: 138,174461616
 TORSW TORSCH SPHLW H R SPH

Einzelschrittausführung eines Programms

Die Funktion eines Programms ist einfacher zu verstehen, wenn sie es Schritt für Schritt laufen lassen und die Wirkung jedes einzelnen Schrittes verfolgen. Diese Vorgehensweise hilft Ihnen, Ihre eigenen Programme zu testen und Fehler zu beseitigen oder von anderen geschriebene Programme zu verstehen.

Die Operationen für die Einzelschrittausführung eines Programms sind im Menü PRG CTRL enthalten.

Einzelschrittoperationen

Tasten	Programmierbarer Befehl	Beschreibung
[↩] [CONT]	CONT	Führt mit der Ausführung eines unterbrochenen Programms fort.
[PRG] CTRL :		
[DEBUG]		Verwendet das Programm oder den Programmnamen in Ebene 1 als Argument. Startet den Programmablauf und unterbricht ihn dann so, als ob der erste Befehl des Programms der Befehl HALT wäre.

Einzelschrittoperationen (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
SST		Führt das nächste Objekt oder den nächsten Befehl im unterbrochenen Programm aus.
SST↓		Hat dieselbe Wirkung wie SST mit einer Besonderheit: wenn der nächste Programmschritt ein Unterprogramm aufruft, wird nur der Sprung in dieses Unterprogramm und nicht das gesamte Unterprogramm ausgeführt.
NEXT		Zeigt das nächste oder die beiden nächsten Objekte an, führt sie aber nicht aus.
HALT	HALT	Unterbricht den Programmablauf an der Stelle, an der der Befehl HALT im Programm steht.
KILL	KILL	Beendet alle unterbrochenen Programme.

Einzelschrittausführung beginnend am Anfang des Programms

In vielen Fällen wollen Sie die Einzelschrittausführung vom Anfang des Programms aus starten. Der allgemeine Ablauf sieht dann folgendermaßen aus:

1. Schreiben Sie das Programm oder den Programmnamen in die Befehlszeile oder in Ebene 1.
2. Drücken Sie **[PRG]** **[CTRL]** **[DEBUG]**. Der Programmablauf wird gestartet und dann abgebrochen, bevor das erste Objekt oder der erste Befehl ausgeführt werden kann. Im Statusbereich wird der HALT-Indikator angezeigt.
3. Auf Wunsch: Drücken Sie **[NEXT]**, um den nächsten oder die nächsten beiden Programmschritte im Statusbereich anzuzeigen,

aber nicht auszuführen. Die Anzeige bleibt bestehen, bis die nächste Taste gedrückt wird.

4. Drücken Sie `SST` einmal, um den ersten Programmschritt im Statusbereich anzuzeigen und dann ausführen zu lassen.
5. Sie haben jetzt mehrere Möglichkeiten:
 - Halten Sie `SST` gedrückt, um die folgenden Programmschritte anzuzeigen und ausführen zu lassen.
 - Drücken Sie `NEXT` zu einem beliebigen Zeitpunkt, um den nächsten oder die nächsten beiden Programmschritte anzuzeigen, aber nicht ausführen zu lassen.
 - Drücken Sie `↶` `CONT`, um den normalen Programmablauf fortzusetzen.
 - Drücken Sie `KILL`, wenn Sie auf den weiteren Programmablauf verzichten wollen.

Beispiel: Einzelschrittausführung eines Programms. Führen Sie das Programm *TORSV* Schritt für Schritt aus. Verwenden Sie den Torus aus dem vorigen Beispiel ($a = 6$ cm, $b = 8$ cm).

Wählen Sie das Menü `VAR` und geben Sie die Daten ein. Übergeben Sie den Programmnamen in die Befehlszeile. Wählen Sie das Menü `PRG CTRL` und führen Sie `DEBUG` aus. Der `HALT`-Indikator erscheint im Statusbereich und zeigt an, daß der Programmablauf begonnen hat und dann abgebrochen wurde.

```
VAR
6 ENTER 8 ENTER
TORSV
PRG CTRL DEBUG
```

```

      HALT
{ HOME }
4:
3:
2:                                     6
1:                                     8
DEBUG SST SST+ NEXT HALT KILL
```

Drücken Sie `SST`. Der erste Programmschritt wird im Statusbereich angezeigt und anschließend ausgeführt.

```
SST
```

```

→ a b
4:
3:
2:
1:
DEBUG SST SST+ NEXT HALT KILL
```

Sie sehen, daß die beiden Argumente im ersten Programmschritt aus dem Stack genommen und in den lokalen Variablen *a* und *b* gespeichert wurden. Am Anfang dieses Abschnitts ist das Vorgehen in 5 Schritten beschrieben. Sie haben die ersten vier Schritte ausgeführt und können nun zwischen den vier Möglichkeiten in Schritt 5 wählen. Für dieses Beispiel können Sie die Einzelschrittausführung fortsetzen, bis der HALT-Indikator verschwindet. Beobachten Sie den Stack und den Statusbereich, während Sie die einzelnen Schritte des Programms nacheinander ausführen lassen.

`SST ... SST`

1:	138,174461616
DEBUG	SST SST↓ NEXT HALT KILL

Einzelschrittausführung einer beliebigen Stelle im Programm

Es kann vorkommen, daß Sie die Einzelschrittausführung an irgendeiner Stelle im Programm, aber nicht mit dem ersten Programmschritt beginnen wollen.

1. Fügen Sie den Befehl HALT an der Stelle in das Programm ein, an der Sie die Einzelschrittausführung starten wollen.
2. Lassen Sie das Programm ablaufen. Bei Ausführung des HALT-Befehls wird das Programm unterbrochen und der HALT-Indikator angezeigt.
3. Führen Sie Schritte 3—5 auf Seite 522 aus.
4. Wenn Sie das Programm wieder normal ablaufen lassen wollen, müssen Sie den Befehl HALT aus dem Programm entfernen.

Einzelschrittausführung von Unterprogrammen

`SST` führt den nächsten Schritt in einem Programm aus. Wenn der nächste Schritt ein Unterprogramm ist, führt `SST` dieses gesamte Unterprogramm in einem Schritt aus. Im vorherigen Beispiel haben Sie mit `SST` das Unterprogramm *TORSA* in einem Schritt ablaufen lassen. Es kann allerdings vorkommen, daß Sie ein Unterprogramm Schritt für Schritt und nicht das Programm als Ganzes ausführen lassen wollen. Dazu können Sie die Operation `SST↓` verwenden. `SST↓` hat ansonsten dieselbe Funktion wie `SST`. Wenn jedoch der nächste Programmschritt ein Unterprogramm ist, springen Sie mit `SST↓` zum ersten Schritt im Unterprogramm. Dort wird wieder unterbrochen.

Beispiel: Einzelschrittausführung eines Unterprogramms. Führen Sie die einzelnen Schritte des Programms *TORSV* aus, um das Volumen eines Torus mit den Radien $a = 10$ cm und $b = 20$ cm zu berechnen. Wenn Sie das Unterprogramm *TORSA* erreichen, führen Sie die einzelnen Befehle Schritt für Schritt aus.

Wählen Sie das Menü VAR und geben Sie die Daten ein. Übergeben Sie den Programmnamen in die Befehlszeile, wählen Sie das Menü PRG CTRL und drücken Sie DEBUG. Lassen Sie die ersten vier Schritte im Programm ausführen und überprüfen Sie dann den nächsten Schritt.

```

[VAR]
10 [ENTER] 12
[ ] TORSV
[PRG] CTRL DEBUG
SST↓ (or SST ) 4 times
NEXT

```

```

TORSV b
4:
3:
2:                               10
1:                               12
[DEBUG] [SST] [SST↓] [NEXT] [HALT] [KILL]

```

Der nächste Schritt ist *TORSA*. Wenn Sie jetzt `SST` drücken, läuft *TORSA* ab. Da Sie *TORSA* in *Einzelschritten* ablaufen lassen wollen, müssen sie `SST↓` drücken. Kontrollieren Sie, ob Sie sich jetzt am ersten Programmschritt von *TORSA*, und nicht am nächsten Programmschritt von *TORSV* befinden.

```

SST↓
NEXT

```

```

→ a
4:
3:
2:                               10
1:                               12
[DEBUG] [SST] [SST↓] [NEXT] [HALT] [KILL]

```

Drücken Sie `SST` oder `SST↓` wiederholt, um die die übrigen Programmteile schrittweise ablaufen zu lassen. Sie können auch zu einem beliebigen Zeitpunkt `[←][CONT]` drücken, um den normalen Programmablauf fortzusetzen.

Tests und Bedingte Strukturen



In diesem Kapitel werden Befehle und Programmstrukturen beschrieben, bei deren gemeinsamer Ausführung Programme Bedingungen abfragen und Entscheidungen treffen.

- *Vergleichsfunktionen* und *Logische Funktionen* veranlassen ein Programm, festzustellen, ob eine bestimmte Bedingung erfüllt ist oder nicht.
- *Bedingte Strukturen* sind Programmstrukturen, die die Ergebnisse solcher Tests verwenden, um Entscheidungen zu treffen.

Beispiel: Tests und bedingte Strukturen. Das Programm in diesem Beispiel verwendet einen Test innerhalb einer bedingten Struktur, um folgende Aufgabe auszuführen:

“Wenn die beiden Zahlen im Stack denselben Wert haben, lösche die eine Zahl aus dem Stack und speichere die andere in der Variablen V1. Wenn die Zahlen allerdings nicht gleich sind, speichere die Zahl von Ebene 1 in V1 und die Zahl von Ebene 2 in V2.”

Programm:	Kommentar:
«	Beginnt das Programm.
DUP2	Kopiert die Zahlen in den Ebenen 1 und 2 in die Ebenen 3 und 4.
IF	Leitet den Testausdruck der bedingten Struktur ein.
SAME	Testet, ob die Zahlen den gleichen Wert haben.
THEN	Beendet den Testausdruck und leitet die Anweisung ein, die ausgeführt wird, wenn das Testergebnis <i>“wahr”</i> ist.
DROP 'V1' STO	Wenn das Testergebnis <i>“wahr”</i> ist (wenn die Zahlen gleich sind), wird eine der Zahlen im Stack gelöscht und die andere Zahl in <i>V1</i> gespeichert.
ELSE	Leitet die Anweisung ein, die ausgeführt wird, wenn das Testergebnis <i>“falsch”</i> ist.
'V1' STO 'V2' STO	Wenn das Testergebnis <i>“falsch”</i> ist (wenn die Zahlen nicht gleich sind), wird die Zahl in Ebene 1 in <i>V1</i> und die Zahl in Ebene 2 in <i>V2</i> gespeichert.
END	Beendet die bedingte Struktur.
»	Beendet das Programm.
[ENTER] [] TST [STO]	Legt das Programm im Stack ab und speichert es in <i>TST</i> .

Geben Sie die Zahlen 26 und 52 ein und führen Sie anschließend *TST* aus, um die beiden Zahlenwerte miteinander zu vergleichen.

26 **[ENTER]** 52
[VAR] TST

H	R	SPH			
---	---	-----	--	--	--

Da die beiden Zahlen nicht gleich waren, enthält das Menü VAR nun zwei neue Variablen $V1$ und $V2$. Durch Drücken der beiden Menütasten können Sie kontrollieren, ob die Variablen die beiden Zahlen enthalten, die Sie eingegeben haben.

Programmtests

Ein Test ist ein algebraisches Objekt oder eine Befehlsfolge, die ein *Testergebnis* im Stack ablegt. Ein Testergebnis ist entweder eine 1 — was bedeutet, daß das Testergebnis *“wahr”* ist oder eine 0 — was bedeutet, daß das Testergebnis *“falsch”* ist. Ein Test kann zum Beispiel der Vergleich $'X < Y'$ sein. Derselbe Vergleich könnte auch als Befehlsfolge ausgeführt werden: $X \ Y \ <$. In beiden Fällen gilt: Wenn X 5 und Y 10 enthält, dann ist das Ergebnis des Tests *“wahr”*, und eine 1 wird im Stack abgelegt. Bedingte Strukturen verwenden ein Testergebnis dazu, zu bestimmen, welche Anweisung der Struktur auszuführen ist. (Bedingte Strukturen werden später in diesem Kapitel besprochen.)

Die Ausdrücke in diesen Tests können in die folgenden Klassen eingeteilt werden:

- *Vergleichsfunktionen.*
- *Logische Funktionen.*
- Befehle, die *Flags testen*. Flags und Befehle, die Flags testen, werden in Kapitel 28 *“Flags”* behandelt.

Diese Befehle befinden sich im Menü PRG TEST (Drücken Sie **PRG** **TEST**).

Vergleichsfunktionen

Vergleichsfunktionen vergleichen zwei Objekte.

Vergleichsfunktionen

Tasten	Programmierbarer Befehl	Beschreibung
PRG TEST (Seiten 1 und 2):		
<code><</code>	<code><</code>	Kleiner als.
<code>></code>	<code>></code>	Größer als.
<code>≤</code>	<code>≤</code>	Kleiner als oder gleich.
<code>≥</code>	<code>≥</code>	Größer als oder gleich.
<code>==</code>	<code>==</code>	Testet die Gleichheit von zwei Objekten. Liefert bei algebraischen Objekten oder Namen einen Ausdruck, dessen Zahlenwert bestimmt werden kann, um ein Testergebnis auf der Grundlage von Zahlenwerten zu erhalten.
<code>≠</code>	<code>≠</code>	Ungleich. Wie <code>==</code> , liefert aber das entgegengesetzte Testergebnis zurück.
<code>SAME</code>	<code>SAME</code>	Wie <code>==</code> , läßt aber keinen Vergleich zwischen dem numerischen Wert eines algebraischen Objektes (oder eines Namens) und einer Zahl zu.

`<`, `>`, `≤` und `≥` vergleichen zwei reelle Zahlen, zwei Binärzahlen oder zwei Zeichenketten und liefern als Vergleichsergebnis entweder 1 ("wahr") oder 0 ("falsch"). Ein Vergleich läuft in der Folge *Ebene 2 - Test - Ebene 1* ab, wobei *Test* die Vergleichsfunktion ist. Wenn zum Beispiel eine 6 in *X* gespeichert ist, bewirkt der Vergleich `× 5 <`, daß 6 und 5 aus dem Stack genommen werden und 0 dort abgelegt wird. Wenn ein Objekt ein algebraischer Ausdruck (oder ein Name) ist und das andere Objekt ein algebraischer Ausdruck (oder Name) oder eine Zahl ist, liefern `<`, `>`, `≤`, und `≥` einen Ausdruck, dessen Zahlenwert bestimmt

werden muß, um ein Testergebnis zu erhalten. Bei Zeichenketten ist "kleiner als" gleichbedeutend mit "weiter vorne" in der alphabetischen Reihenfolge. Zum Beispiel ist "AAA" kleiner als "AAB".

`==` Nimmt zwei Objekte aus dem Stack und macht dann folgendes:

- Wenn beide Objekte weder ein algebraischer Ausdruck noch ein Name sind, wird, falls beide Objekte vom selben Typ sind und denselben Wert haben, eine `1` geliefert, im anderen Fall eine `0`. Bei Listen und Programmen wird davon ausgegangen, daß sie denselben Wert haben, wenn die in ihnen enthaltenen Objekte identisch sind.
- Wenn ein Objekt ein algebraischer Ausdruck (oder ein Name) ist und das andere Objekt ein algebraischer Ausdruck (oder Name) oder eine Zahl ist, wird ein Ausdruck geliefert, dessen Zahlenwert bestimmt werden muß, um ein Testergebnis zu erhalten.

(Beachten Sie bitte, daß `==` für Vergleichsoperationen verwendet wird, während `=` die beiden Seiten einer Gleichung trennt.) `≠` Arbeitet wie `=`, liefert jedoch das entgegengesetzte Testergebnis.

`SAME` liefert eine `1` ("wahr"), wenn zwei Objekte identisch sind. Zum Beispiel wird bei `'X+3'` `4` `SAME` eine `0` geliefert, unabhängig vom Wert der Größe `X`, da der algebraische Ausdruck `'X+3'` nicht mit der reellen Zahl `4` identisch ist. Für alle anderen Objekttypen außer algebraischen Objekten und Namen bewirkt `SAME` genau dasselbe wie `==`.

Anwendung von Vergleichsfunktionen in algebraischen Ausdrücken. Mit Ausnahme von `SAME` können Vergleichsfunktionen in algebraischen Ausdrücken als *Infix*-Funktionen, d.h. zwischen den Operanden, angewendet werden. Ist zum Beispiel die Zahl `6` in `X` gespeichert, sendet `'X<5'` `NUM` eine `0` zurück.

Logische Funktionen

Logische Funktionen liefern ein Testergebnis, das vom Ergebnis zweier bereits ausgeführter Tests ausgeht. Beachten Sie, daß diese vier Funktionen *jedes von null verschiedene Argument* als das Ergebnis "wahr" interpretieren.

Logische Funktionen

Tasten	Programmierbarer Befehl	Beschreibung
PRG TEST (Seite 1):		
AND	AND	Liefert 1 ("wahr"), wenn beide Argumente "wahr" sind.
OR	OR	Liefert 1 ("wahr"), wenn ein Argument "wahr" ist oder beide Argumente "wahr" sind.
XOR	XOR	Liefert 1 ("wahr"), wenn ein Argument "wahr" ist, aber nicht beide.
NOT	NOT	Liefert 1 ("wahr"), wenn das Argument 0 ("falsch") ist; im anderen Fall liefert NOT eine 0 ("falsch").

AND, OR und XOR werden verwendet, um zwei Prüfergebnisse zu verknüpfen. Ist zum Beispiel 4 in Y gespeichert, liefert die Operation $Y \ 8 < 5 \ \text{AND}$ eine 1. Zuerst legt $Y \ 8 <$ eine 1 zurück im Stack ab. AND nimmt 1 und 5 vom Stack, interpretiert beide Zahlen als wahre Ergebnisse und legt eine 1 im Stack ab.

NOT sendet die logische Umkehrung eines Testergebnisses zurück. Wenn zum Beispiel 1 in X gespeichert und 2 in Y gespeichert ist, liefert $Y < \text{NOT}$ eine 0.

Anwendung logischer Funktionen in algebraischen Ausdrücken.
AND, OR und XOR können in algebraischen Ausdrücken als *Infix*-Funktionen angewendet werden. '3<5 XOR 4>7' →NUM zum Beispiel liefert eine 1.

NOT kann in algebraischen Ausdrücken als *Prefix*-Funktion angewendet werden. Zum Beispiel liefert 'NOT Z≤4' →NUM eine 0, wenn Z = 2 ist.

Objekttypen testen

Der Befehl TYPE (**PRG** **TEST** **TYPE**) akzeptiert ein beliebiges Objekt als Argument und liefert die Zahl, die diesen Objekttyp kennzeichnet. Die Tabelle auf Seite 102 in Kapitel 4 gibt eine Übersicht über die Objekte des HP 48 und die zugehörigen Typnummern.

Bedingte Strukturen

Die *bedingten Strukturen* des HP 48 veranlassen ein Programm dazu, als Folge eines oder mehrerer Tests eine Entscheidung zu treffen. Bedingte Strukturen bestehen aus Befehlen, die nur wirksam sind, wenn sie in der richtigen Weise miteinander kombiniert werden. Diese Befehle sind im Menü PRG BRCH enthalten (**PRG** **BRCH**).

Die bedingten Strukturen sind :

- IF...THEN...END.
- IF...THEN...ELSE...END.
- CASE...END.

Die Struktur IF...THEN...END

IF...THEN...END führt eine Folge von Befehlen nur dann aus, wenn ein Test ein Ergebnis liefert, das "wahr" ist. Die Syntax lautet:

IF *Testausdruck* THEN *Anweisung* END

Der Testausdruck kann eine Befehlsfolge (zum Beispiel $A \ B \ \neq$) oder ein algebraischer Ausdruck (zum Beispiel $'A \leq B'$) sein. Ist die Prüfungsweise ein algebraischer Ausdruck, wird dessen Zahlenwert *automatisch bestimmt*. (\rightarrow NUM oder EVAL ist nicht erforderlich).

Sie können  IF als Schreibhilfe verwenden, um das Gerüst der Verzweigung einzugeben:

```
IF
THEN
END
```

Beispiel 1: IF...THEN...END. Die beiden folgenden Programme prüfen den Wert in Ebene 1. Ist der Wert positiv, wird das Vorzeichen umgekehrt. Das erste Programm verwendet eine Befehlsfolge als Testausdruck:

```
« DUP IF 0 > THEN NEG END »
```

Der Wert im Stack muß dupliziert werden, weil der Befehl $>$ zwei Argumente aus dem Stack nimmt (die mit DUP erzeugte Kopie des Wertes und 0).

Die nächste Version verwendet einen algebraischen Ausdruck als Testausdruck:

```
«  $\rightarrow$  x « IF ' $x > 0$ ' THEN x NEG END » »
```

Beispiel 2: IF...THEN...END. Dieses Programm multipliziert zwei Zahlen miteinander, wenn sie ungleich Null sind.

Programm:

Kommentar:

```
«
```

```
 $\rightarrow$  x y
```

Erzeugt die lokalen Variablen x und y , die die beiden Zahlen vom Stack enthalten.

«

IF	Beginnt den Testausdruck.
'x≠0'	Testet eine Zahl und legt ein Testergebnis im Stack ab.
'y≠0'	Testet die andere Zahl und legt ein weiteres Testergebnis im Stack ab.
AND	Stellt fest, ob beide Prüfungen das Ergebnis "wahr" geliefert haben.
THEN	Beendet den Testausdruck und leitet die Anweisung ein, die ausgeführt wird, wenn das Testergebnis "wahr" ist.
x y *	Wenn AND das Ergebnis "wahr" liefert, werden mit dieser Operation die beiden Zahlen miteinander multipliziert.
END	Beendet die Prüfanweisung.

»

»

Das folgende Programm führt die gleiche Aufgabe aus wie das vorhergehende Programm:

```
« → x y IF 'x AND y' THEN x y * END »
```

Der Testausdruck 'x AND y' liefert das Ergebnis "wahr", wenn beide Zahlen ungleich Null sind.

Wie IF...THEN...END wirkt. IF leitet den Testausdruck ein, der ein Testergebnis im Stack ablegt. THEN nimmt das Testergebnis aus dem Stack. Wenn der Wert ungleich Null ist, wird die auf THEN folgende Anweisung ausgeführt. Im anderen Fall wird der Programmablauf nach END fortgesetzt.

Die Struktur IF...THEN...ELSE...END

IF...THEN...ELSE...END löst eine bestimmte Folge von Befehlen aus, wenn ein Test das Ergebnis "wahr" liefert und eine andere Befehlsfolge, wenn der Test das Ergebnis "falsch" liefert. Die Syntax lautet:

```
IF Testausdruck THEN Anweisung ELSE Anweisung END
```

Ist der Testausdruck ein algebraischer Ausdruck, wird *automatisch* dessen Zahlenwert *bestimmt* (\rightarrow NUM oder EVAL ist nicht erforderlich).

Sie können  IF  als Schreibhilfe verwenden, um das Gerüst der Verzweigung einzugeben:

```
IF
THEN
ELSE
END
```

Beispiel 1: IF...THEN...ELSE...END. Das folgende Programm nimmt einen Wert x aus dem Stack und berechnet die Funktion $\sin x/x$. Bei $x = 0$ liefert die Division einen Fehler, so daß das Programm in diesem Fall den Grenzwert 1 ausgibt:

```
⊗ → x ⊗ IF 'x≠0' THEN x SIN x / ELSE 1 END ⊗ ⊗
```

Beispiel 2: IF...THEN...ELSE...END. Dieses Programm multipliziert wie das Beispiel 2 für IF...THEN...END zwei Zahlen miteinander, wenn sie beide von Null verschieden sind. Zusätzlich liefert das Programm aber die Zeichenkette "ZERO", wenn einer der beiden Werte 0 ist.

Programm:

Kommentar:

```
⊗
```

```
→ n1 n2
```

Speichert die Werte aus den Ebenen 1 und 2 in lokalen Variablen.

```
⊗
```

Beginnt die definierende Prozedur der Struktur der lokalen Variablen.

IF	Beginnt den Testausdruck.
'n1≠0 AND n2≠0'	Testet <i>n1</i> und <i>n2</i> .
THEN	Wenn beide Zahlen ungleich Null sind ...
n1 n2 *	... werden die beiden Werte multipliziert.
ELSE	Wenn beide Werte nicht ungleich Null sind ...
"ZERO"	... wird die Zeichenkette ZERO ausgegeben.
END	Beendet die bedingte Struktur.
⌘	Beendet die definierende Prozedur.
⌘	

Wie IF...THEN...ELSE...END wirkt. IF leitet den Testausdruck ein, der ein Testergebnis im Stack ablegt. THEN nimmt das Prüfergebnis vom Stack. Wenn der Wert ungleich Null ist, wird die auf THEN folgende Anweisung ausgeführt. Im anderen Fall läuft die auf ELSE folgende Anweisung ab. Nach Beendigung der entsprechenden Anweisung wird der Programmablauf nach END fortgesetzt.

Die Struktur CASE...END

Die Struktur CASE...END erlaubt die Unterscheidung einer Reihe von Fällen. Der erste Test, der das Ergebnis "wahr" liefert, veranlaßt die Ausführung der entsprechenden Anweisung und beendet die Struktur CASE...END. Zusätzlich können Sie nach dem letzten Test eine Standardanweisung einfügen, die ausgeführt wird, wenn alle Tests das Ergebnis "falsch" liefern.

Die Struktur CASE...END hat folgende Syntax:

```
CASE
    Testausdruck1 THEN Anweisung1 END
    Testausdruck2 THEN Anweisung2 END
    :
    Testausdruckn THEN Anweisungn END
    Standardanweisung (bei Bedarf)
END
```

Auch hier existieren zwei Schreibhilfen für das Gerüst der Verzweigung und zwar  CASE für:

```
CASE
THEN
END
END
```

und  CASE für:

```
THEN
END
```

Beispiel: Die Struktur CASE...END. Das folgende Programm speichert das Argument in Ebene 1 in einer Variablen, wenn das Argument eine Zeichenkette, eine Liste oder ein Programm ist.

Programm:

```
«
→ y
«
CASE
```

Kommentar:

Speichert das Argument in der lokalen Variablen y.
Beginnt die definierende Prozedur
Beginnt die Case-Struktur.

y TYPE 2 SAME THEN y 'STR' STO END	Fall 1: Ist das Argument eine Zeichenkette, wird diese in <i>STR</i> gespeichert.
y TYPE 5 SAME THEN y 'LIST' STO END	Fall 2: Ist das Argument eine Liste, wird diese in <i>LIST</i> gespeichert.
y TYPE 8 SAME THEN y 'PROG' STO END	Fall 3: Ist das Argument ein Programm, wird dieses in <i>PROG</i> gespeichert.
END	Beendet die Case-Struktur.
»	Beendet die definierende Prozedur.
»	

Wie CASE...END wirkt. Bei der Ausführung von CASE wird der Testausdruck₁ ausgewertet. Liefert dieser Test das Ergebnis "wahr", folgt die Ausführung der Anweisung₁ und dann ein Sprung nach END. Liefert der Testausdruck₁ das Ergebnis "falsch", wird der Programmablauf mit Testausdruck₂ fortgesetzt. Der Programmablauf innerhalb der CASE-Struktur wird solange fortgesetzt, bis eine Anweisung ausgeführt wird oder bis alle Testausdrücke abgearbeitet, falls sie alle das Ergebnis "falsch" liefern. Bei Bedarf kann eine Standardanweisung eingefügt werden. Die Standardanweisung wird dann ausgeführt, wenn alle Testausdrücke das Ergebnis "falsch" liefern.

Bedingte Befehle

Die Anwendung der Strukturen IF...THEN...END und IF...THEN...ELSE ist in solchen Situationen hilfreich, in denen die Anweisungen aus *mehreren* Befehlen und Objekten bestehen. Mit den beiden Befehlen IFT (Wenn...dann) und IFTE (Wenn...dann...sonst) können Sie leicht den gleichen Entscheidungsprozeß ablaufen lassen, wenn die bedingten Anweisungen nur aus einem *einzelnen* Befehl oder Objekt bestehen.

Der Befehl IFT (Wenn-dann-Ende)

Der Befehl IFT verwendet zwei Argumente: Ein Testergebnis in Ebene 2 und ein Objekt in Ebene 1. Das Objekt in Ebene 1 wird ausgeführt, wenn der Test das Ergebnis "wahr" liefert.

Beispiel: Der Befehl IFT. Das folgende Programm nimmt eine Zahl vom Stack und zeigt POSITIVE an, wenn die Zahl positiv ist.

```
« 0 > "POSITIVE" IFT »
```

Die Funktion IFTE

Die Funktion IFTE verwendet drei Argumente: Ein Testergebnis in 3 und Objekte in den Ebenen 2 und 1. Das Objekt in Ebene 2 (die "Wahr-Anweisung") wird ausgeführt, wenn die Prüfung das Ergebnis "wahr" liefert. Im anderen Fall wird das Objekt in Ebene 1 ausgeführt.

Beispiel: Der Befehl IFTE. Dieses Programm holt einen Wert aus Ebene 1 und zeigt POSITIVE an, wenn die Zahl positiv oder gleich Null ist, im anderen Fall NEGATIVE:

```
« 0 ≥ "POSITIVE" "NEGATIVE" IFTE »
```

Anwendung von IFTE in algebraischen Ausdrücken. Die Funktion IFTE kann auch als Funktion in algebraischen Ausdrücken verwendet werden. Sie hat die Syntax:

```
IFTE(Testausdruck, Anweisung, Anweisung)
```

Beispiel: Die Funktion IFTE. Dieses Programm stellt eine benutzerdefinierte Funktion dar, die eine Zahl (x) vom Stack holt und $\sin(x)/x$ berechnet, falls x ungleich Null ist. Ist $x = 0$, liefert das Programm eine 1:

```
« → x 'IFTE(x≠0, SIN(x)/x, 1)' »
```

Aufbau von Programmschleifen



Programmschleifen lassen einen Programmteil mehrere Male hintereinander ablaufen. Man unterscheidet zwei wichtige Arten von Programmschleifen:

- Bei einer *bestimmten Schleife* legt das Programm vorher fest, wie oft die Schleifenanweisung auszuführen ist.
- In einer *unbestimmten Schleife* testet das Programm bei jedem Durchlauf, ob die Schleife ein weiteres Mal durchlaufen werden soll.

Wie die bedingten Strukturen, die in Kapitel 26 beschrieben wurden, sind Programmschleifen aus Befehlen aufgebaut, die nur in der richtigen Kombination wirksam sind. Diese Befehle sind im Menü PRG BRCH enthalten (**PRG** **BRCH**).

Aufbau von bestimmten Programmschleifen

Man unterscheidet zwei verschiedene Strukturen bei bestimmten Programmschleifen, und von jeder gibt es nochmals zwei Varianten:

- START...NEXT und START...STEP.
- FOR...NEXT und FOR...STEP.

Die Struktur START...NEXT

START...NEXT führt einen Programmteil mehrfach aus; dabei ist festgelegt, wie oft die Programmschleife durchlaufen wird. Die Syntax lautet:

Anfangswert Endwert START *Schleifenanweisung* NEXT

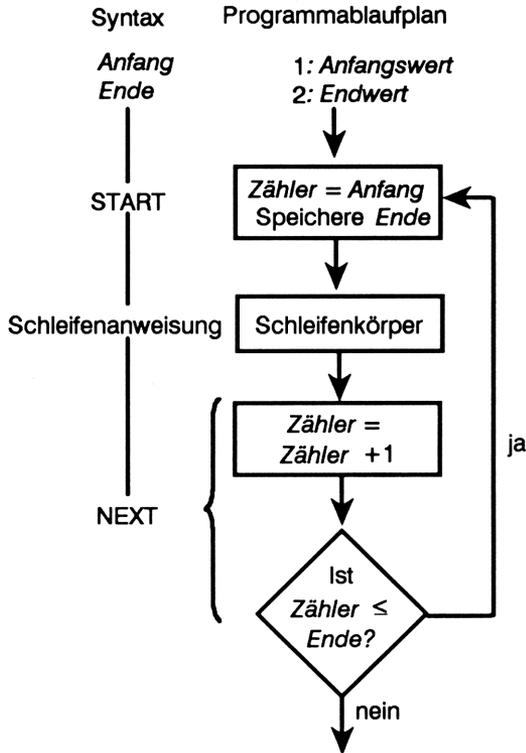
Sie können  START als Schreibhilfe verwenden, um das Gerüst der Schleife einzugeben:

```
START  
NEXT
```

Beispiel: Eine START...NEXT-Schleife. Das folgende Programm legt eine Liste an, die zehn Kopien der Zeichenkette "ABC" enthält:

```
« 1 10 START "ABC" NEXT 10 →LIST »
```

Wie START...NEXT arbeitet. START nimmt zwei Zahlen (*Anfangswert* und *Endwert*) vom Stack und speichert sie als Anfangs- und Endwerte für einen Schleifenzähler. Anschließend wird die Schleifenanweisung ausgeführt. NEXT erhöht den Zähler um 1 und prüft, ob sein Wert kleiner als der oder gleich dem *Endwert* ist. Wenn ja, wird die Schleifenanweisung noch einmal ausgeführt.



Beachten Sie, daß die Schleifenanweisung mindestens einmal ausgeführt wird.

Die Struktur START...STEP

START...STEP bewirkt das gleiche wie START...NEXT, jedoch sind hier andere Werte als 1 zur Erhöhung des Schleifenzählers zugelassen. Die Syntax lautet:

Anfangswert Endwert START *Schleifenanweisung* *Schrittweite* STEP

Sie können  START als Schreibhilfe verwenden, um das Gerüst der Schleife einzugeben:

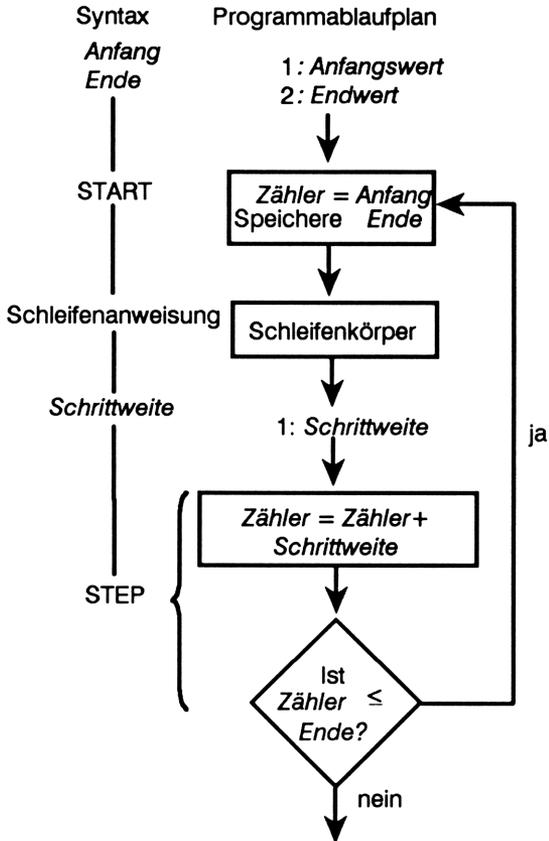
```
START
STEP
```

Beispiel: Eine START...STEP-Schleife. Das folgende Programm verwendet eine Zahl x vom Stack und berechnet das Quadrat dieser Zahl $x/3$ mal:

```
« DUP → x « x 1 START x SQ -3 STEP »
```

Wie START...STEP arbeitet. START verwendet zwei Zahlen (*Anfangswert* und *Endwert*) vom Stack und speichert sie als Anfangs- und Endwerte für den Schleifenzähler. Anschließend wird die Schleifenanweisung ausgeführt. STEP nimmt die Schrittweite vom Stack und erhöht den Zähler um diesen Wert. Wenn das Argument von STEP ein algebraischer Ausdruck oder ein Name ist, wird automatisch dessen Zahlenwert bestimmt.

Die Schrittweite kann positiv oder negativ sein. Ist sie positiv, wird die Schleife noch einmal durchlaufen, wenn der Wert des Zählers kleiner als oder gleich dem *Endwert* ist. Ist die Schrittweite negativ, wird die Schleife noch einmal durchlaufen, wenn der Wert des Zählers größer als oder gleich dem *Endwert* ist. Im folgenden Ablaufdiagramm ist die Schrittweite positiv.



Die Struktur FOR...NEXT

Eine FOR...NEXT-Schleife führt ein Programmsegment mehrfach aus; dabei ist genau festgelegt, wie oft die Programmschleife durchlaufen wird. Als Schleifenzähler dient eine lokale Variable. Sie können diese Variable innerhalb der Schleife verwenden. Die Syntax lautet:

Anfangswert *Endwert* FOR *Zähler* *Schleifenanweisung* NEXT

Sie können  FOR als Schreibhilfe verwenden, um das Gerüst der Schleife einzugeben:

```
FOR
NEXT
```

Beispiel 1: Eine FOR...NEXT-Schleife. Das folgende Programm überträgt die Quadrate der ganzen Zahlen von 1 bis 5 in den Stack:

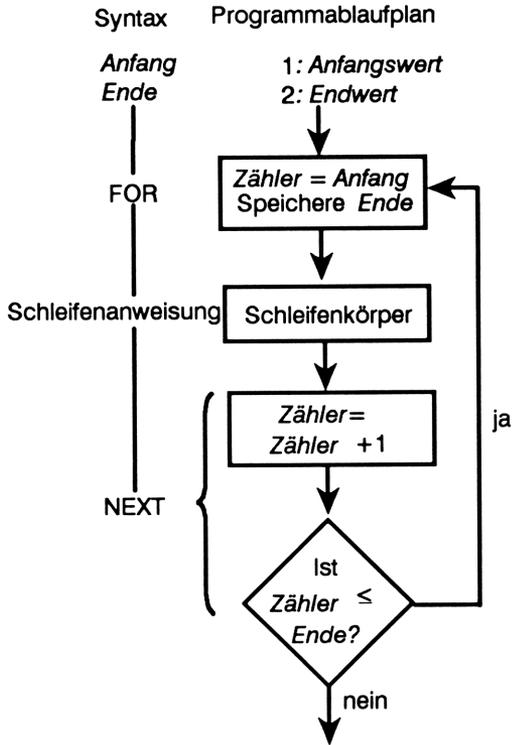
```
« 1 5 FOR j j SQ NEXT »
```

Beispiel 2: Eine FOR...NEXT-Schleife. Das folgende Programm holt den Wert x vom Stack und berechnet die ganzzahligen Potenzen i von x . Ist zum Beispiel $x = 12$, der *Anfangswert* ist 3 und der *Endwert* ist 5, dann liefert das Programm die Werte 12^3 , 12^4 und 12^5 . Als Eingabedaten benötigt die Schleife *Anfangswert* und *Endwert* in den Ebenen 3 und 2 sowie x in Ebene 1:

```
« → x « FOR n 'x^n' EVAL NEXT » »
```

→ x entfernt x vom Stack, während *Anfangswert* und *Endwert* als Argument für FOR auf dem Stack bleiben.

Wie FOR...NEXT arbeitet. FOR nimmt *Anfangswert* und *Endwert* vom Stack, verwendet sie als Anfangs- und Endwerte für den Schleifenzähler und erzeugt anschließend die lokale Variable *Zähler* als Schleifenzähler. Danach wird die Schleifenanweisung ausgeführt; *Zähler* kann innerhalb der Schleifenanweisung erscheinen. NEXT erhöht *Zähler* um eins und prüft anschließend, ob *Zähler* kleiner als der oder gleich dem *Endwert* ist. Wenn ja, wird die *Schleifenanweisung* wiederholt (mit dem neuen Wert für *Zähler*). Beim Verlassen der Schleife wird *Zähler* gelöscht.



Die Struktur FOR...STEP

FOR...STEP bewirkt das gleiche wie FOR...NEXT, jedoch sind hier andere Werte als 1 für die Erhöhung des Schleifenzählers zugelassen. Die Syntax lautet:

Anfangswert Endwert FOR *Zähler Schleifenanweisung Schrittweite* STEP

Sie können  FOR als Schreibhilfe verwenden, um das Gerüst der Schleife einzugeben:

```
FOR
STEP
```

Beispiel 1: Eine FOR...STEP-Schleife. Das folgende Programm überträgt die Quadrate der ganzen Zahlen 1, 3, 5, 7 und 9 in den Stack:

```
« 1 9 FOR x x SQ 2 STEP »
```

Beispiel 2: Eine FOR...STEP-Schleife. Das folgende Programm verwendet n vom Stack und überträgt die Zahlenfolge 1, 2, 4, 8, 16, ... n in den Stack. Wenn n nicht in der Folge enthalten ist, hält das Programm bei dem letzten Wert an, der kleiner ist als n :

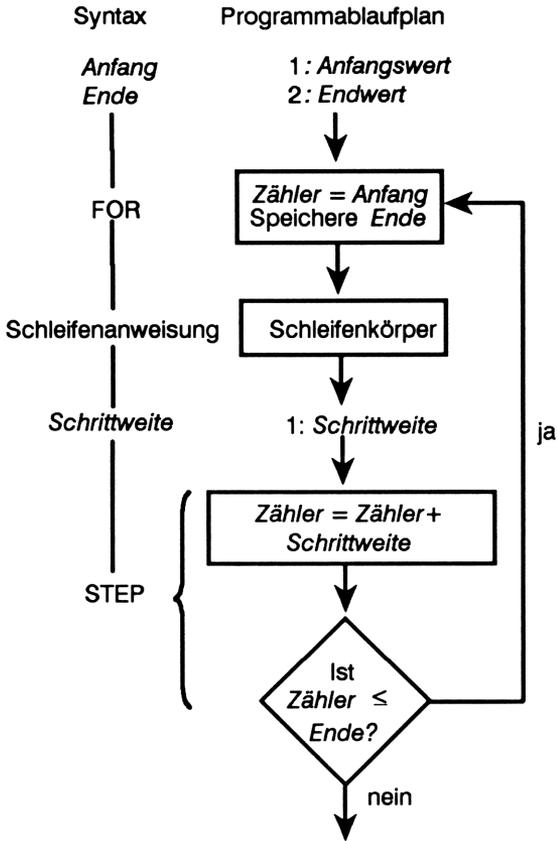
```
« 1 SWAP FOR n n n STEP »
```

Wie FOR...STEP arbeitet. FOR nimmt *Anfangswert* und *Endwert* vom Stack, verwendet sie als Anfangs- und Endwerte für den Schleifenzähler und erzeugt anschließend die lokale Variable *Zähler* als Schleifenzähler. Als nächstes wird die Schleifenanweisung ausgeführt; *Zähler* kann innerhalb der Schleifenanweisung erscheinen. STEP holt die Schrittweite vom Stack und erhöht *Zähler* um diesen Wert.

Die Schrittweite kann positiv oder negativ sein. Ist sie positiv, wird die Schleife noch einmal durchlaufen, wenn der Wert von *Zähler* kleiner als oder gleich dem *Endwert* ist. Ist die Schrittweite negativ, wird die Schleife noch einmal durchlaufen, wenn der Wert von *Zähler* größer als oder gleich dem *Endwert* ist.

Beim Verlassen der Schleife wird *Zähler* gelöscht.

Im folgenden Ablaufdiagramm ist die Schrittweite positiv.



Aufbau von unbestimmten Programmschleifen

Die Struktur DO...UNTIL...END

DO...UNTIL...END... führt eine Schleife mehrere Male aus, bis ein Test das Ergebnis wahr (ein Ergebnis ungleich Null) liefert. Da der *Testausdruck* nach der Schleifenanweisung ausgewertet wird, wird die Schleife immer mindestens einmal durchlaufen. Die Syntax lautet:

```
DO Schleifenanweisung UNTIL Testausdruck END
```

Sie können  als Schreibhilfe verwenden, um das Gerüst der Schleife einzugeben:

```
DO
UNTIL
END
```

Beispiel: Eine DO...UNTIL...END-Schleife. Das folgende Programm berechnet $n + 2n + 3n + \dots$ für einen Wert n . Das Programm wird abgebrochen, wenn die Summe den Wert 1000 überschreitet; die Summe und der Koeffizient von n werden im Stack abgelegt.

Programm:

```
«
DUP 1 → n s c
«
DO
```

Kommentar:

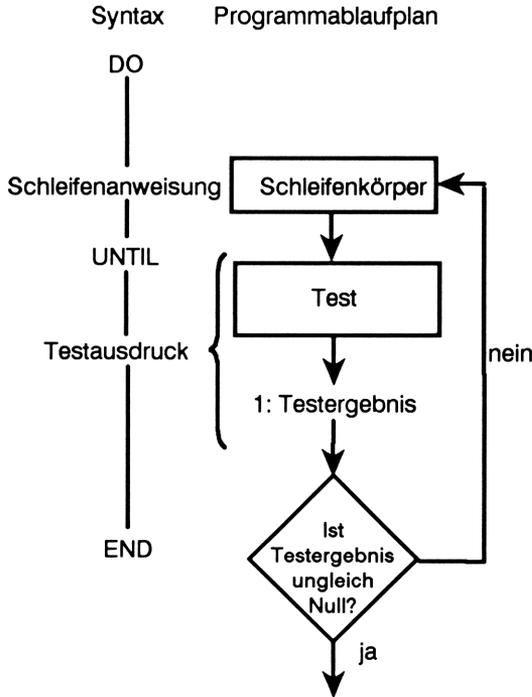
Dupliziert n und speichert den Wert in n und s ; Stellt den Zähler auf den Anfangswert $c = 1$.

Beginnt die definierende Prozedur, in diesem Fall ein Programm, für die Struktur lokaler Variablen.

Beginnt die Schleifenanweisung.

'c' INCR	Erhöht den Zähler um 1. (INCR wird auf Seite 553 behandelt.)
n * 's' STO+	Berechnet $c \times n$ und addiert das Produkt zu s .
UNTIL	Beginnt den Testausdruck
≤ 1000 >	Wiederholt den Schleifendurchlauf bis $s > 1000$.
END	Beendet den Testausdruck.
≤ c	Überträgt s und c auf den Stack.
»	Beendet die definierende Prozedur.
»	

Wie DO...UNTIL...END arbeitet. DO beginnt die Ausführung der Schleifenanweisung. UNTIL beendet die Schleifenanweisung und beginnt den Testausdruck. Der Testausdruck liefert ein Testergebnis in den Stack. END nimmt das Testergebnis vom Stack. Ist sein Wert Null, wird die Schleifenanweisung noch einmal ausgeführt; im anderen Fall wird der Programmablauf nach END fortgesetzt.



Die Struktur WHILE...REPEAT...END

WHILE...REPEAT...END führt wiederholt einen Test aus und durchläuft dann eine Schleifenanweisung, wenn der Test das Ergebnis wahr liefert. Da der Testausdruck vor der Schleifenanweisung ausgewertet wird, wird die Schleifenanweisung, wenn die Prüfung bereits am Anfang ein falsches Ergebnis liefert, überhaupt nicht ausgeführt. Die Syntax lautet:

WHILE *Testausdruck* REPEAT *Schleifenanweisung* END

Sie können  WHILE als Schreibhilfe verwenden, um das Gerüst der Schleife einzugeben:

```
WHILE  
REPEAT  
END
```

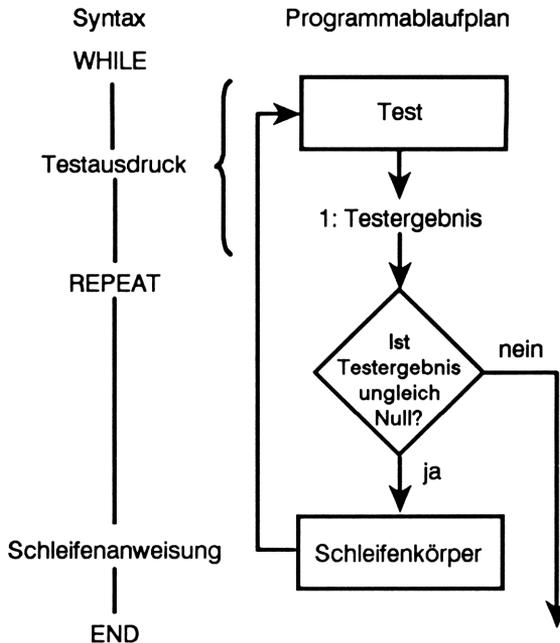
Beispiel 1: Eine WHILE...REPEAT...END-Schleife. Das folgende Programm dividiert eine Zahl, die zu Beginn im Stack steht, mehrmals durch 2; diese Operation wird solange wiederholt, wie das Ergebnis noch ganzzahlig teilbar ist. Beginnt das Programm zum Beispiel mit der Zahl 24, so berechnet es die Zahlen 12, 6 und 3:

```
« WHILE DUP 2 MOD 0 == REPEAT 2 / DUP END DROP »
```

Beispiel 2: Eine WHILE...REPEAT...END-Schleife. Das folgende Programm nimmt eine beliebige Zahl von Vektoren oder Feldern vom Stack und addiert sie zur Statistikmatrix. (Die Vektoren und Felder müssen alle dieselbe Spaltenzahl haben.) WHILE...REPEAT...END wird anstelle von DO...UNTIL...END verwendet, weil der Test *vor* der Addition ausgeführt werden muß. (Befinden sich *nur* Vektoren oder Felder mit derselben Spaltenzahl im Stack, gibt das Programm eine Fehlermeldung aus, nachdem der letzte Vektor oder das letzte Feld zur Statistikmatrix addiert wurde.)

```
« WHILE DUP TYPE 3 == REPEAT Σ+ END »
```

Wie WHILE...REPEAT...END arbeitet. Der Testausdruck wird ausgewertet und überträgt ein Testergebnis in den Stack. REPEAT nimmt den Wert vom Stack. Ist der Wert ungleich Null, wird die Schleifenanweisung ausgeführt; im anderen Fall wird das Programm nach END fortgesetzt.



Schleifenzähler (INCR und DECR)

Der Befehl INCR (vorwärts zählen) (`[MEMORY] INCR`) benötigt den Namen einer globalen oder einer lokalen Variablen als Argument. Die Variable muß eine reelle Zahl enthalten. Der Befehl

- Erhöht den in der Variablen gespeicherten Wert um 1.
- Gibt den neuen Wert der Variablen aus.

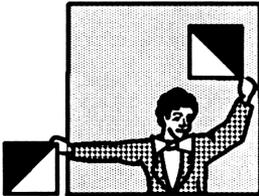
Enthält c zum Beispiel den Wert 5, sendet 'c' INCR eine 6 in den Stack zurück und speichert die 6 in c .

Der Befehl DECR (rückwärts zählen) arbeitet wie INCR, verringert jedoch den Wert der Variablen.

Beispiel: Verwendung eines Schleifenzählers mit einer unbestimmten Schleife. Das folgende Programm verwendet maximal fünf Vektoren vom Stack und addiert sie zur aktuellen Statistikmatrix.

Programm:	Kommentar:
«	
0 → c	Speichert 0 in der lokalen Variablen <i>c</i> .
«	Beginnt die definierende Prozedur für die Struktur lokaler Variablen.
WHILE	Beginnt den Testausdruck
DUP TYPE 3 ==	Liefert das Ergebnis wahr, wenn Ebene 1 einen Vektor enthält.
'c' INCR	Erhöht den Wert in <i>c</i> und überträgt den erhöhten Wert in Ebene 1.
5 ≤	Liefert das Ergebnis wahr, wenn der erhöhte Wert $c \leq 5$ ist.
AND	Liefert das Ergebnis wahr, wenn die beiden letzten Tests beide das Ergebnis wahr geliefert haben.
REPEAT	
Σ+	Addiert den Vektor zu ΣDAT .
END	Beendet die WHILE...REPEAT-Schleife.
»	Beendet die definierende Prozedur.
»	

Flags



Flags sind ein wichtiges Hilfsmittel bei der Programmierung des HP 48. Man kann sich ein Flag (Zustandsmarke) als einen Schalter vorstellen, der entweder eingeschaltet (*gesetzt*) oder ausgeschaltet (*rückgesetzt*) ist. Ein Programm kann den Zustand eines Flags in einer bedingten Verzweigung oder einer Programmschleife überprüfen, um eine Entscheidung zu treffen. (Bedingte Verzweigungen und Programmschleifen wurden in den vorhergehenden Kapiteln behandelt). Das Testen von Flags erweitert die Fähigkeiten des Taschenrechners, Entscheidungen zu treffen, über die Möglichkeiten, die Vergleichs- und Logikfunktionen bieten, hinaus.

Flag-Typen

Der HP 48 verwendet zwei verschiedene Flag-Typen: *System-Flags*, die Nummern von -1 bis einschließlich -64 tragen, und *Benutzer-Flags* mit Nummern von 1 bis 64 . System-Flags haben eine festgelegte Bedeutung für den Taschenrechner. Zum Beispiel beeinflusst das System-Flag -40 die Anzeige der Uhr — wenn dieses Flag *rückgesetzt* ist (der Standardzustand), wird die Uhr nur bei der Wahl des Menüs TIME angezeigt. Ist dieses Flag *gesetzt*, wird die Uhr ständig angezeigt. (Wenn Sie im Menü MODES \square CLK \square drücken, setzen oder löschen Sie das Flag -40 .) In Anhang E sind die 64 System-Flags und deren Definitionen aufgelistet.

Eingebaute Operationen verwenden keine Benutzer-Flags; deren Bedeutung hängt ausschließlich davon ab, wie *Sie* sie definieren. Wenn Sie ein Benutzer-Flag von 1 bis einschließlich 5 setzen, wird der entsprechende Indikator aktiviert. (Beachten Sie bitte, daß Einsteckkarten die Einstellungen der Benutzer-Flags 31 — 64 beeinflussen können. Einsteckkarten werden in Kapitel 34 behandelt.)

Setzen, Rücksetzen und Testen von Flags

Die folgenden Befehle verwenden eine Flag-Nummer als Argument — eine ganze Zahl von 1 bis 64 (für Benutzer-Flags) oder -1 bis -64 (für System-Flags).

Flag-Befehle

Tasten	Programmierbarer Befehl	Beschreibung
[PRG] TEST (Seite 3) (oder [MODES] Seiten 2 und 3):		
[SF]	SF	Setzt das Flag.
[CF]	CF	Setzt das Flag zurück.
[FS?]	FS?	Liefert 1 (wahr), wenn das Flag gesetzt ist oder 0, wenn es rückgesetzt ist.
[FC?]	FC?	Liefert 1 (wahr), wenn das Flag rückgesetzt ist oder 0, wenn es gesetzt ist.
[FS?C]	FS?C	Testet das Flag (liefert 1, wenn das Flag gesetzt ist) und setzt das Flag anschließend zurück.
[FC?C]	FC?C	Testet das Flag (liefert 1, wenn das Flag nicht gesetzt ist) und setzt das Flag anschließend zurück.

Beispiel: Testen eines System-Flags. Das folgende Programm setzt einen Termin auf den 6. Juni 1991, 17.05 Uhr fest. Es testet zuerst mit Hilfe einer bedingten Verzweigung den Zustand des System-Flags -42 (das Flag für das Datumsformat) und liefert dann in Abhängigkeit vom Testergebnis das Datum des Termins im aktuell gültigen Datumsformat.

Programm:

Kommentar:

«

IF
-42 FC?

Prüft den Zustand des Flags -42, des Flags für das Datumsformat.

THEN
6.151991

Ist das Flag -42 rückgesetzt, liefert das Programm das Datum im Format *month/day/year* (Monat/Tag/Jahr).

ELSE
15.061991

Ist das Flag -42 gesetzt, liefert das Programm das Datum im Format *day.month.year* (Tag/Monat/Jahr).

END

Beendet die Verzweigungsstruktur.

17.05 "TEST COMPLETE"
3 →LIST STOALARM

Schließt die Befehlsfolge zur Einstellung des Termins ab. (17.05 ist die Uhrzeit des Termins und "TEST COMPLETE" ist die bei Fälligkeit des Termins angezeigte Meldung.)

»

Beispiel: Benutzer-Flags in Programmen. Das folgende Programm liefert in Abhängigkeit vom Zustand des Benutzer-Flags 10 entweder den Nachkommateil oder den ganzzahligen Teil des Argumentes in Ebene 1.

Programm:**Kommentar:**

«

IF	Beginnt die Verzweigungsstruktur
10 FS?	Testet den Zustand des Benutzer-Flags 10.
THEN	Ist Flag 10 gesetzt ...
IP	... wird der ganzzahlige Teil ausgegeben.
ELSE	Ist Flag 10 rückgesetzt ...
FP	... wird der Nachkommateil ausgegeben.
END	Beendet die Verzweigungsstruktur

»

Bevor Sie dieses Programm laufen lassen, müssen Sie Flag 10 setzen, wenn Sie den ganzzahligen Teil des Argumentes benötigen; brauchen Sie den Nachkommateil des Argumentes, so müssen Sie Flag 10 rücksetzen. Flag 10 hat also eine ganz bestimmte Bedeutung in diesem Programm: sein Zustand bestimmt, welcher Teil des Argumentes in Ebene 1 in den Stack zurückgegeben wird.

Aufrufen und Speichern der Flag-Zustände

Die Befehle RCLF (Flag-Status aufrufen) und STOF (Flag-Status speichern) erlauben es, die Flag-Zustände des HP 48 aufzurufen und anschließend zu speichern. Die Befehle erlauben es einem Programm, das den Zustand eines mehrerer Flags während seines Ablaufs verändert, den ursprünglichen Zustand wiederherzustellen.

Aufrufen der Flag-Zustände

RCLF liefert eine Liste mit zwei 64-Bit-Binärzahlen, die den aktuellen Status aller System-Flags beziehungsweise aller Benutzer-Flags darstellen:

$$\{ \#n_s, \#n_u \}$$

Die niederwertigsten Bits (die am weitesten rechts stehenden Binärstellen) von $\#n_s$ und $\#n_u$ geben die Zustände des System-Flags -1 beziehungsweise des Benutzer-Flags $+1$ wieder.

Speichern der Flag-Zustände

STOF legt die Zustände entweder nur der System-Flags oder die Zustände der System- und der Benutzer-Flags fest. Der Befehl verwendet als Argument entweder:

- Eine einzelne Binärzahl ($\#n_s$), wobei dann nur die System-Flags in die entsprechenden Zustände gebracht werden.
- Eine Liste mit zwei Binärzahlen ($\{ \#n_s, \#n_u \}$), wobei dann System- und Benutzer-Flags in die entsprechenden Zustände gebracht werden.

Ein Bit mit dem Wert 1 setzt das entsprechende Flag; ein Bit mit dem Wert 0 setzt es zurück. Die niederwertigsten Bits (die am weitesten rechts stehenden Binärstellen) von $\#n_s$ und $\#n_u$ legen die Zustände des System-Flags -1 beziehungsweise des Benutzer-Flags $+1$ fest.

Das Programm *PRESERVE* auf Seite 601 in Kapitel 31 verwendet die Befehle RCLF und STOF.

Dialogprogramme



Einfache Programme wie die in Kapitel 25 verwenden Daten, die vor dem Ablauf des Programms bereitgestellt werden, und sie liefern als Ergebnisse nackte Zahlen. Solche Programme sind oft schwierig anzuwenden; besonders für jemanden, der das Programm nicht selbst geschrieben hat. Der Benutzer muß wissen, welche Argumente im Stack abzulegen sind und in welcher Reihenfolge sie einzugeben sind, und er muß wissen, wie die vom Programm in den Stack geschriebenen Ergebnisse zu interpretieren sind.

Interaktive Programme (Dialogprogramme) dagegen unterstützen den Benutzer in folgender Weise:

- Sie unterbrechen den Programmablauf und fordern zur Eingabe von Daten auf.
- Sie zeigen Programmergebnisse mit Erläuterungen oder Markierungen an.
- Sie unterbrechen den Programmablauf und überlassen dem Benutzer die Entscheidung, wie das Programm fortgesetzt werden soll.

Unterbrechung des Programmablaufs für eine Dateneingabe

Befehle zur Dateneingabe

Tasten	Programmierbarer Befehl	Beschreibung
 CONT	CONT	Setzt ein unterbrochenes Programm fort.
PRG CTRL (Seiten 1, 2 und 3):		
 HALT	HALT	Hält einen Programmablauf an.
 INPUT	INPUT	Unterbricht den Programmablauf für eine Dateneingabe. Verhindert Stackoperationen, während das Programm angehalten ist.
 PROM	PROMPT	Hält einen Programmablauf für eine Dateneingabe an.
 DISP	DISP	Stellt ein Objekt in der angegebenen Zeile des Anzeigefeldes dar.
 WAIT	WAIT	Unterbricht den Programmablauf für x Sekunden, wobei x eine Zahl in Ebene 1 ist.
 KEY	KEY	Liefert ein Testergebnis nach Ebene 1; liefert außerdem, falls eine Taste gedrückt wird, deren Position.
 BEEP	BEEP	Erzeugt ein akustisches Signal einer bestimmten Frequenz und einer bestimmten Dauer.

Befehle zur Dateneingabe (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
[PRG] DSPL (Seite 4):		
CLLCD FREEZ	CLLCD FREEZE	Löscht das Anzeigefeld. "Friert" einen bestimmten Bereich des Anzeigefeldes ein. Dieser Bereich bleibt dann solange unverändert, bis eine Taste gedrückt wird.

Der Befehl PROMPT

PROMPT nimmt eine Zeichenkette als Argument aus Ebene 1, stellt die Zeichenkette (ohne die Begrenzungszeichen ") im Statusbereich dar und hält den Programmablauf an. Der Taschenrechner kann jetzt wieder über die Tastatur bedient werden. Die Ausführung des Programms wird durch Drücken von CONT fortgesetzt. Wenn Sie zum Beispiel das Programmsegment

« "ABC?" PROMPT »

ablaufen lassen, erscheint folgende Anzeige:

ABC?

4:
3:
2:
1:

PARTS PROB HYP MATR WECTR GMBE

Diese Anzeige bleibt solange erhalten, bis Sie **[ENTER]** oder **[ATTN]** drücken oder bis Sie den Statusbereich aktualisieren (zum Beispiel durch Drücken von **[←] [REVIEW]**).

Das folgende Programm *TPROMPT* fordert Sie auf, die Abmessungen eines Torus einzugeben, ruft anschließend das Programm *TORSA* (Kapitel 25, Seite 519) auf, um seine Oberfläche zu berechnen. Sie brauchen vor der Ausführung des Programms die Daten nicht in den Stack zu schreiben.

Argumente	Ergebnisse
1:	1: Fläche

Programm:

Kommentar:

«

"ENTER a, b IN ORDER:"

Legt die Zeichenkette zur Bedienungsführung im Stack ab.

PROMPT

Stellt die Zeichenkette im Statusbereich dar, unterbricht den Programmablauf und gibt die Bedienung des Taschenrechners über die Tastatur wieder frei.

TORSA

Führt das Programm *TORSA* aus und verwendet dabei die gerade in den Stack eingegebenen Argumente.

»

[ENTER] [] TPROMPT [STO]

Speichert das Programm in *TPROMPT*.

Beispiel: Aufforderung zur Dateneingabe in einem Programm.

Lassen Sie das Programm *TPROMPT* laufen, um das Volumen eines Torus mit dem Innenradius $a = 8$ cm und dem Außenradius $b = 10$ cm zu berechnen.

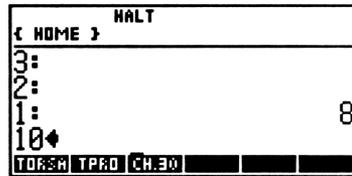
Wählen Sie das Menü VAR, und starten Sie *TPROMPT*.

[VAR] TPROMPT

ENTER a, b IN ORDER:
 4:
 3:
 2:
 1:
 TORSA TPROMPT CH.EM

Das Programm fordert Daten von Ihnen. Geben Sie den inneren und den äußeren Radius ein. Beachten Sie, daß die Aufforderung zur Dateneingabe im Statusbereich nach dem Drücken von **[ENTER]** gelöscht wird.

8 **[ENTER]** 10



Setzen Sie das Programm fort.

[←] **[CONT]**



Das Ergebnis wird in Ebene 1 im Stack abgelegt.

Beachten Sie, daß Sie nach der Programmunterbrechung durch PROMPT Operationen mit dem Taschenrechner wie vor Beginn des Programmablaufs ausführen können. Nehmen Sie an, der Außenradius b des Torus im vorigen Beispiel wurde mit 3,94 Zoll gemessen. Sie können diesen Wert in cm umwandeln, während das Programm für die Dateneingabe unterbrochen ist, indem Sie 3,94 **[ENTER]** 2,54 **[x]** drücken.

Der Befehl BEEP

Mit dem Befehl BEEP können akustische Signale programmiert und so die Möglichkeit der Bedienungserweiterung erweitert werden. BEEP verwendet zwei Argumente vom Stack: Die Tonfrequenz von Ebene 2 und die Tondauer von Ebene 1. In der folgenden, erweiterten Version von *TPROMPT* werden Sie durch einen 440-Hz-Ton mit einer Dauer von einer halben Sekunde zur Eingabe von Daten aufgefordert.

Programm:

Kommentar:

❖

```
"ENTER a, b IN ORDER:"
```

```
440 .5 BEEP
```

Löst ein Tonsignal aus, um die Aufforderung zur Eingabe von Daten akustisch zu unterstützen.

```
PROMPT
```

```
TORSA
```

❖

Die Befehle DISP, HALT und FREEZE

DISP, HALT und FREEZE können zusammen verwendet werden, um einen Bediener zur Dateneingabe aufzufordern:

- DISP stellt ein Objekt in eine bestimmte Zeile des Anzeigefeldes. DISP benötigt zwei Argumente vom Stack: Das Objekt in Ebene 2 und in Ebene 1 eine Zeilennummer zwischen 1 und 7 für das Anzeigefeld von 1 bis 7. Um die Anzeige von Meldungen zu erleichtern, stellt DISP *Zeichenketten*-Objekte ohne die einschließenden Begrenzungszeichen " dar.

Beachten Sie, daß die mit DISP dargestellte Anzeige nur solange erhalten bleibt, wie das Programm läuft. Wenn das Programm endet oder wenn es durch den Befehl HALT unterbrochen wird, kehrt der Taschenrechner in die normale Stackumgebung zurück, und es erscheint automatisch eine Anzeige der aktuellen Daten.

- FREEZE "friert" einen oder mehrere Anzeigebereiche ein, so daß sie erst wieder nach einem *Tastendruck* aktuelle Daten anzeigen. Das Argument *n* in Ebene 1 ist die Summe der Codes für die Bereiche, die eingefroren werden sollen. Die Codes lauten: 1 für den Statusbereich, 2 für den Stack-/Befehlszeilen-Bereich, 4 für den Menübereich.

- HALT unterbricht den Programmablauf an der Stelle, an der der Befehl HALT gegeben wird und aktiviert den HALT-Indikator. Sie können jetzt über die Tastatur normale Operationen mit dem Taschenrechner durchführen. Der Programmablauf wird durch Drücken von CONT (oder SST) fortgesetzt.

Beim Ausführen des Programms

```
« "ABC■DEF■GHI" CLLCD 1 DISP 3 FREEZE HALT »
```

erscheint die folgende Anzeige:



(Das ■ in diesem Programm ist die Darstellung des Taschenrechners für das Zeichen ◀ (neue Zeile), sobald ein Programm auf den Stack übertragen worden ist.)

Der Befehl INPUT

INPUT wird für die Aufforderung zur Dateneingabe verwendet, wenn der Programmierer verhindern will, daß der Anwender Stackoperationen durchführt. Betrachten Sie das folgende Programm:

```
« "Variable name?" ":VAR:" INPUT »
```

Wenn dieses Program ausgeführt wird, erscheint folgende Darstellung im Anzeigefeld:

```

{ HOME } PRG
-----
Variable name?

:VAR: ←
PRTS PRG HYP MATR VECTR BASE

```

1. Der Stackbereich wird gelöscht und die Zeichenkette aus Ebene 2, Variable name?, wird oben im Stackbereich angezeigt. Sie wird als *Aufforderungs-Zeichenkette* bezeichnet.
2. Die Zeichenkette aus Ebene 1, :VAR:, wird in der Befehlszeile angezeigt. Sie wird die *Befehlszeilen-Zeichenkette* genannt. Der Programmeingabemodus wird aktiviert und der Einfüge-Cursor hinter die Zeichenkette gesetzt. Das Programm ist jetzt für die Dateneingabe unterbrochen.
3. Die Programmausführung wird durch Drücken von **ENTER** fortgesetzt, wodurch der Inhalt der Befehlszeile als Zeichenkette im Stack abgelegt wird. Sie wird *Ergebnis-Zeichenkette* genannt.

Das folgende Programm *VSPH* bestimmt das Volumen einer Kugel. *VSPH* berechnet zuerst $\frac{4}{3}\pi$ und fordert Sie anschließend auf, den Radius der Kugel einzugeben. Danach führt es die Berechnung durch. Da sich ein Teilergebnis der Berechnung bereits im Stack befindet, schützt *VSPH* den Stack, indem es die Eingabe des Radius mit INPUT anfordert. INPUT ruft den Programmeingabemodus auf, wenn der Programmablauf für die Dateneingabe angehalten ist. Die folgenden Befehle werden nicht sofort ausgeführt — stattdessen erscheinen sie in der Befehlszeile, bis der Bediener **ENTER** drückt.

Argumente	Ergebnisse
1:	1: <i>Volumen</i>

Programm:

Kommentar:

⌘

4 3 / π * →NUM

Beginnt mit der Berechnung.

"Key in radius"

Erzeugt die Zeichenkette zur Aufforderung an den Bediener, die oben im Stackbereich angezeigt wird.

""

Erzeugt die Zeichenkette für die Befehlszeile. In diesem Fall ist die Zeichenkette leer, so daß in der Befehlszeile keine Anzeige erscheint.

INPUT

Stellt den Bedienerhinweis im Stackbereich dar, setzt den Cursor an den Anfang der Befehlszeile und unterbricht das Programm für die Eingabe der Daten (des Kugelradius).

OBJ→

Wandelt die Ergebnis-Zeichenkette in eine reelle Zahl um.

3 ^ *

Bildet die dritte Potenz des Radius und beendet die Berechnung.

⌘

[ENTER] [] VSPH [STO]

Speichert das Programm in *VSPH*.

Beispiel: Aufforderung zur Dateneingabe mit INPUT. Lassen Sie *VSPH* laufen, um das Volumen einer Kugel mit dem Radius 2,5 Meter zu berechnen.

Wählen Sie das Menü VAR, und starten Sie das Programm.

[VAR] VSPH



Drücken Sie  **[DROP]**, um zu kontrollieren, wie INPUT den Stack schützt.

 **[DROP]**

```
PRG
{ HOME }
Key in radius
DROP
WSPH TDRSH TPRO CH30
```

In der Befehlszeile wird DROP angezeigt, aber nicht ausgeführt, so daß das Teilergebnis in Ebene 1 stehen bleibt.

Drücken Sie **[ATTN]**, um die Befehlszeile wieder zu aktivieren. Geben Sie anschließend den Radius ein, und setzen Sie den Programmablauf fort.

[ATTN]
2.5 **[ENTER]**

```
1: 65,4498469497
WSPH TPRO TDRSH
```

Wahlmöglichkeiten für den Befehl INPUT. In seiner allgemeinen Form ist das Argument für INPUT in Ebene 1 eine *Liste*, die Inhalt und Interpretation der Befehlszeile bestimmt. Die Liste kann *einen oder mehrere* der folgenden Parameter *in beliebiger Reihenfolge* enthalten:

- Die *Befehlszeilen-Zeichenkette*, deren Inhalt zur Bedienungsführung bei unterbrochenem Programm in der Befehlszeile angezeigt wird.
- Entweder eine *reelle Zahl* oder eine *Liste, mit zwei reellen Zahlen*, mit der oder mit denen die Anfangsposition des Cursors in der Befehlszeile festgelegt wird:
 - Eine reelle Zahl n bedeutet, daß der Cursor auf das n te Zeichen vom linken Ende der ersten Zeile in der Befehlszeile gesetzt wird. Eine *positive* Zahl n wählt den Einfüge-Cursor; eine *negative* Zahl n wählt den Überschreibe-Cursor. 0 bestimmt das Ende einer Zeichenkette in der Befehlszeile.
 - Eine Liste, mit der die Zeile und die Spalte der Anfangsposition des Cursors festgelegt wird: Die erste Zahl legt eine Zeile innerhalb der Befehlszeile fest (1 bedeutet die erste Zeile der Befehlszeile); die zweite Zahl gibt die Anzahl der Zeichen vom linken Rand der festgelegten Zeile an. 0 bestimmt das Ende einer Zeichenkette in der angegebenen Zeile der Befehlszeile. Eine *positive* Zeilenzahl wählt den Einfüge-Cursor an; eine *negative* Zeilenzahl wählt den Überschreibe-Cursor.

- Einen oder mehrere der Parameter ALG, α oder V, die als Namen ohne Anführungszeichen einzugeben sind:
 - ALG aktiviert den algebraischen Programmeingabe-Modus.
 - α (α \rightarrow A) verriegelt die alphabetische Tastatur.
 - V prüft, ob die Zeichen in der Ergebnis-Zeichenkette, ohne die Begrenzungszeichen " , ein gültiges Objekt oder gültige Objekte bilden. Wenn diese Zeichen kein gültiges Objekt oder keine gültigen Objekte bilden, gibt INPUT die Warnung `Invalid Syntax` im Anzeigefeld aus und fordert noch einmal zur Eingabe von Daten auf.

Der Standardzustand von INPUT. Sie können sich bei der Wahl der Parameter für die Liste in Ebene 1 auf einen einzigen beschränken. Die nicht gewählten Parameter nehmen dann folgende Standardzustände an:

- Die Befehlszeile ist leer.
- Der Einfüge-Cursor befindet sich am Ende der Befehlszeilen-Zeichenkette.
- Programmeingabe-Modus.
- Die Ergebnis-Zeichenkette wird nicht auf ungültige Syntax geprüft.

Wenn Sie *nur* eine Befehlszeilen-Zeichenkette als Argument in Ebene 1 angeben, brauchen Sie sie nicht in Form einer Liste einzugeben. Zum Beispiel legt das vorher behandelte Programm *VSPH* eine leere Zeichenkette in der Befehlszeile fest.

Aufbau einer Befehlszeilen-Zeichenkette. Nachdem der Bediener seine Daten in die Befehlszeile eingegeben und **[ENTER]** gedrückt hat, um die Ausführung des Programms fortzusetzen, wird der Inhalt der Befehlszeile als Ergebnis-Zeichenkette in Ebene 1 abgelegt. Um die eingegebenen Daten zu verarbeiten, kann das Programm an einer bestimmten Stelle `OBJ→` ausführen, um die Ergebnis-Zeichenkette in *ein gültiges Objekt* oder *gültige Objekte* umzuwandeln. Dazu legt das Programm für die Befehlszeilen-Zeichenkette eine bestimmte Form fest und führt anschließend, wenn nach der Eingabe die Ergebnis-Zeichenkette in Ebene 1 steht, geeignete Operationen aus:

- Das Programm kann für die Befehlszeile eine *leere* Zeichenkette festlegen. In diesem Fall besteht die Ergebnis-Zeichenkette *nur* aus den Eingabedaten. Das Programm *VSPH* auf Seite 568 wendet diese Methode an.

- Das Programm kann eine Zeichenkette für die Befehlszeile festlegen, deren Zeichen die Markierungs- und Begrenzungszeichen für ein *markiertes Objekt* bilden. (Markierte Objekte werden ausführlich auf Seite 92 besprochen.) In diesem Fall vervollständigt die Eingabe das markierte Objekt. Das Programm *TINPUT* auf Seite 572 wendet diese Methode an.
- Das Programm kann eine Zeichenkette für die Befehlszeile festlegen, deren Zeichen eine Meldung bilden. In einem solchen Fall entfernt das Programm diese Zeichen aus der Ergebnis-Zeichenkette, so daß nur die Eingabe in Form einer Zeichenkette übrig bleibt. Das Programm *SSEC* auf Seite 574 wendet diese Methode an.

In den ersten beiden Fällen kann der Parameter *V* auch als Teil des Argumentes in Ebene 1 angegeben werden, um die erneute Aufforderung zur Eingabe von Daten zu veranlassen, wenn die Ergebnis-Zeichenkette keine gültigen Objekte enthält.

Das folgende Programm *TINPUT* wendet *INPUT* an, um den Bediener zur Eingabe des Innen- und des Außenradius eines Torus aufzufordern und ruft anschließend *TORSA* auf (Kapitel 25, Seite 519), um dessen Oberfläche zu berechnen. *TINPUT* fordert die Eingabe der Daten *a* und *b* in einer zweizeiligen Befehlszeile; das Argument für *INPUT* in Ebene 1 ist eine Liste bestehend aus:

- der Befehlszeilen-Zeichenkette.
- einer inneren Liste, die die Anfangsposition des Cursors festlegt.
- den Parameter *V* zur Prüfung, ob die Ergebnis-Zeichenkette einen Syntaxfehler enthält.

Die Befehlszeilen-Zeichenkette bildet die Markierungs- und Begrenzungszeichen für zwei markierte Objekte. In der Liste wird der Eingabemodus nicht festgelegt, so daß der Programmeingabe-Modus gilt (der Standardmodus).

Argumente	Ergebnisse
1:	1: <i>Fläche</i>

Programm:

Kommentar:

«

"Key in a, b"

Erzeugt die Zeichenkette, die oben im Stackbereich angezeigt wird, das Argument in Ebene 2.

{

Beginnt mit dem Argument, der Liste in Ebene 1.

" :a: :b:" {1 0} V

Die Liste enthält die Befehlszeilen-Zeichenkette, eine Liste und den Parameter zur Überprüfung der Syntax. (Um die Zeichenkette einzugeben, müssen Sie  " "  : a   : b drücken. Wenn Sie später **ENTER** drücken, um das beendete Programm auf den Stack zu übertragen, wird die Zeichenkette in einer Zeile angezeigt, wobei das Zeichen ■ für das Zeichen "neue Zeile" steht.) Die innere Liste setzt den Einfüge-Cursor auf die Position direkt nach : a: in Zeile 1. V veranlaßt, daß geprüft wird, ob die Ergebnis-Zeichenkette einen Syntaxfehler enthält.

}

Schließt die Liste ab.

INPUT

Stellt die Zeichenketten im Stackbereich und in der Befehlszeile dar, positioniert den Cursor, wie in der Liste in Ebene 1 festgelegt und wählt, weil nichts anderes angegeben wurde, den Programmeingabemodus; unterbricht anschließend die Programmausführung für die Dateneingabe und prüft, ob die Ergebnis-Zeichenkette Syntaxfehler enthält.

OBJ→

Wandelt die Zeichenkette in ihre einzelnen Objekte um (in diesem Fall in zwei markierte Objekte).

TORSA

Ruft *TORSA* auf, um die Oberfläche zu berechnen.

✳

[ENTER] **[]** TINPUT **[STO]**

Speichert das Programm in *TINPUT*.

Beispiel: Aufforderung zur Dateneingabe mit INPUT. Führen Sie *TINPUT* aus, um die Oberfläche eines Torus mit dem Innenradius $a = 10$ cm und dem Außenradius $b = 20$ cm zu berechnen.

Wählen Sie das Menü VAR, und starten Sie das Programm.

[VAR] TINPU

{ HOME }	PRG
Key in a, b	
:a:↕	
:b:	
TINPU	VSPH TORSA TPRO CH.30

Geben Sie den Wert für a ein und drücken Sie **[▼]**, um den Cursor in die nächste Aufforderungszeile zu setzen. Geben Sie danach den Wert für b ein.

10 **[▼]** 20

{ HOME }	PRG
Key in a, b	
:a:10	
:b:20↕	
TINPU	VSPH TORSA TPRO CH.30

Setzen Sie den Programmablauf fort.

[ENTER]

1:	2960,88132033
TINPU	VSPH TPRO TORSA

Das folgende Program verwendet INPUT für die Aufforderung, eine Sozialversicherungsnummer einzugeben; anschließend wählt es die ersten drei Ziffern und die letzten vier Ziffern aus der Ergebnis-Zeichenkette in Form einer Zeichenkette. Das Argument für INPUT in Ebene 1 legt fest:

- Eine Befehlszeilen-Zeichenkette.
- Den *Überschreibe*-Cursor, der sich am Anfang der Befehlszeilen-Zeichenkette (- 1) befinden soll. Der *Überschreibe*-Cursor veranlaßt den Bediener dazu, die Befehlszeilen-Zeichenkette "auszufüllen", wobei er mit  die Gedankenstriche in der Sozialversicherungsnummer überspringen kann.
- Den Programmeingabe-Modus (Standard).
- Keine Überprüfung der Objektsyntax — die Gedankenstriche in der Sozialversicherungsnummer sind keine gültigen Zeichen außerhalb der Begrenzungszeichen der Zeichenkette (Standard).

Argumente	Ergebnisse
2:	2: " <i>Die ersten drei Ziffern</i> "
1:	1: " <i>Die letzten vier Ziffern</i> "

Programm:

Kommentar:

«

"Key in S.S. #"

Erzeugt die Zeichenkette in Ebene 2, die oben im Stackbereich angezeigt wird.

{ " - - " -1 }

Erzeugt das Argument für INPUT in Ebene 1. (Geben Sie 3 Leerstellen ein zwischen dem ersten Begrenzungszeichen " und dem ersten -, zwei Leerstellen zwischen den beiden - und 4 Leerstellen zwischen dem letzten - und dem Schlußbegrenzungszeichen ".

INPUT

Unterbricht das Programm für die Dateneingabe.

DUP 1 3 SUB
SWAP
8 11 SUB

Kopiert die resultierende Zeichenkette und entnimmt anschließend die ersten drei und die letzten vier Ziffern in Form einer Zeichenkette.

✳

ENTER **↑** SSEC **STO**

Speichert das Programm in *SSEC*.

Kennzeichen der ausgegebenen Daten

Eine erläuternde Kennzeichnung oder eine Meldung kann die Verständlichkeit der von einem Programm ausgegebenen Daten verbessern.

Verwenden von markierten Objekten als ausgegebene Daten

Sie können das von einem Programm gelieferte Ergebnis kennzeichnen, indem Sie den Befehl →TAG verwenden. →TAG (**PRG** **OBJ** →TAG) verwendet zwei Argumente: Ein beliebiges Objekt in Ebene 2 und einen Namen, eine Zeichenkette oder eine reelle Zahl (das Markierungszeichen) in Ebene 1.

Das folgende Programm *TTAG* ist mit *TINPUT* identisch, außer daß es das Ergebnis markiert.

Programm:**Kommentar:**

«

"Key in a, b"

C ":a:=:b:" (1 0) V)

INPUT OBJ→

TORSA

'AREA'

Erzeugt das Markierungszeichen, in diesem Fall einen Namen.

→TAG

Verbindet das Markierungszeichen mit dem Objekt in Ebene 2, dem Ergebnis, und erzeugt so das markierte Objekt.

»

ENTER **□** TTAG **STO**Speichert das Programm in *TTAG*.

Beispiel: Verwenden eines markierten Objektes für die Ausgabe von Daten. Führen Sie *TTAG* aus, um die Fläche eines Torus mit dem Innenradius $a = 1.5$ und $b = 1.85$ zu berechnen.

Wählen Sie das Menü VAR, und starten Sie das Programm. Geben Sie die Werte a und b ein, und setzen Sie die Programmausführung fort. Das Ergebnis wird als markiertes Objekt in den Stack übertragen.

VAR **TTAG**1.5 **▼** 1.85**ENTER**

1:	AREA:	11,5721111603
TTAG	TINPU	WSPH TORSA TP&D

Verwenden von Zeichenketten-Befehlen zur Kennzeichnung ausgegebener Daten

Sie können Zeichenketten-Befehle und DISP dazu verwenden, ein Objekt zu kennzeichnen und anzuzeigen, das in Ebene 1 des Stack steht:

1. Wandeln Sie das Objekt mit →STR (**PRG** **OBJ** **→STR**) in eine Zeichenkette um.
2. Legen Sie eine Zeichenkette für die Kennzeichnung im Stack ab.

3. Vertauschen Sie die beiden Zeichenketten auf dem Stack, und verknüpfen Sie sie anschließend (SWAP +).
4. Zeigen Sie die resultierende Zeichenkette an (*n* DISP).

Das folgende Programm *TSTRING* ist mit *TINPUT* identisch, außer daß es das Ergebnis in eine Zeichenkette umwandelt und ihm eine Zeichenkette zur Kennzeichnung anhängt.

Programm:

Kommentar:

⌘

"Key in a, b"

(":a:~:~:~:~:b:" (1 0) V)

INPUT OBJ→

TORSA

→STR

Wandelt das Ergebnis in eine Zeichenkette um.

"Area = "

Gibt die Zeichenkette zur Kennzeichnung ein.

SWAP +

Vertauscht die Positionen der beiden Zeichenketten im Stack und fügt sie zusammen.

CLLCD 1 DISP 1 FREEZE

Stellt die resultierende Zeichenkette ohne ihre Begrenzungszeichen in Zeile 1 des Anzeigefeldes dar.

⌘

[ENTER] **[]** TSTRING **[STO]**

Speichert das Programm in *TSTRING*.

Beispiel: Kennzeichnung ausgegebener Daten. Führen Sie *TSTRING* aus, um die Fläche des Torus aus dem vorigen Beispiel zu berechnen ($a = 1.5$, $b = 1.85$).

Wählen Sie das Menü VAR, und starten Sie das Programm. Geben Sie die Werte für a und b ein, und setzen Sie die Programmausführung fort. Das gekennzeichnete Ergebnis wird im Statusbereich dargestellt.

```
VAR TSTRI
1.5 ▾ 1.85
ENTER
```

```
Area = 11,5721111603
4:
3:
2:
1:
TSTRI TTMS TINPU VSPH TORSH TPRD
```

Unterbrechen eines Programms, um Daten anzuzeigen

Der Befehl WAIT (**PRG** **CTRL** **NXT** WAIT) unterbricht den Programmablauf für x Sekunden, wobei x eine positive, reelle Zahl in Ebene 1 ist. Sie können WAIT zusammen mit DISP dazu verwenden, während des laufenden Programms Meldungen anzuzeigen — zum Beispiel Zwischenergebnisse.

WAIT interpretiert die Argumente 0 and -1 unterschiedlich — Einzelheiten finden Sie im Abschnitt “Befehle, die eine Tastenadresse liefern” auf Seite 584.

Verwenden von Menüs in Programmen

Anwendungsmenüs wie die Menüs SOLVE und PLOT und auch die Menüs VAR und CST können in Programmen genauso aufgerufen und verwendet werden wie bei der normalen Ausführung von Operationen über die Tastatur.

Anzeige eines internen Menüs

Um ein internes Menü in einem Programm anzuzeigen, müssen Sie den Befehl MENU eingeben (**PRG** **CTRL** **NXT** MENU), zusammen mit der Zahl, die dem internen Menü zugeordnet ist. Die Tabelle in Anhang D enthält alle Menüs des HP 48 und die zugehörigen Menünummern. Zum Beispiel wird mit 20 MENU die Seite 1 des Menüs MODES aufgerufen. Sie können eine bestimmte Seite in einem Menü aktivieren, indem Sie das Argument in der Form $xx.yy$ eingeben, wobei xx die Menünummer und yy die Seitenzahl ist.

Das folgende Programm ruft die dritte Seite im Menü MODES auf und fordert Sie auf, den Winkelmodus einzustellen.

```
◀ 20.03 MENU "Select Angle Mode" PROMPT ▶
```

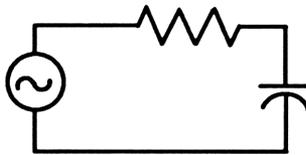
RCLMENU ( **MODES**  RCLM) liefert die Menünummer des zuletzt angezeigten Menüs.

Benutzerspezifische Menüs in Programmen

In Kapitel 15 haben Sie erfahren, wie ein benutzerspezifisches Menü mit einer Liste als Argument für MENU aufgebaut werden kann. In Programmen können Sie benutzerspezifische Menüs für folgende Aufgaben gestalten:

- Nachahmen von internen Anwendungsprogrammen wie HP Solve.
- Aufforderung, bestimmte Entscheidungen zu treffen.

Nachahmen von internen Anwendungsprogrammen. Mit dem folgenden Programm *EIZ* können Sie ein benutzerspezifisches Menü aufbauen, mit dem Sie die Benutzeroberfläche von HP-Solve bei der Berechnung von kapazitiven elektrischen Schaltungen nachahmen können.



Die Anwendung des Ohmschen Gesetzes auf diese Schaltung ergibt folgenden Ausdruck:

$$E = IZ$$

Darin bedeuten:

E ist die Spannung, die an der Schaltung liegt.

I ist der Strom, der durch die Schaltung fließt.

Z ist der komplexe Widerstand der Schaltung.

Da Spannung, Strom und Widerstand komplexe Größen sind, können Sie mit HP-Solve keine Lösungen für diese Gleichung bestimmen. Das kundenspezifische Menü in *EIZ* weist der links umgeschalteten Taste für jede Variable eine *direkte* Lösung zu, und ordnet der nicht umgeschalteten und der rechts umgeschalteten Taste die *Speicher-* und *Aufruf-*Funktionen zu — die Funktion der Tasten ist also analog zu HP-Solve.

Programm:

Kommentar:

«

```
DEG
-15 SF -16 SF 2 FIX
```

Stellt den Gradmodus ein. Setzt die Flags -15 und -16, um die komplexen Zahlen in Polarform darzustellen. Stellt den Anzeigemodus auf 2 Stellen hinter dem Komma ein.

(

Beginnt die Liste für das benutzerspezifische Menü.

```
( "E" ( « 'E' STO »
« I Z * DUP 'E' STO
"E" →TAG
CLLCD 1 DISP
1 FREEZE »
« E » } }
```

Legt die Funktion der Menütaste 1 fest, die mit  bezeichnet wird. Wenn Sie  drücken, wird das Objekt in Ebene 1 in der Variablen *E* gespeichert. Wenn Sie   drücken, wird das Produkt aus *I* und *Z* berechnet, in der Variablen *E* gespeichert und als markiertes Objekt angezeigt. Wenn Sie   drücken, wird das in *E* gespeicherte Objekt nach Ebene 1 ausgegeben.

```
( "I" ( « 'I' STO »
« E Z / DUP 'I' STO
"I" →TAG
CLLCD 1 DISP
1 FREEZE »
« I » } }
```

Legt die Funktion der Menütaste 2 fest.

```
( "Z" ( « 'Z' STO »
« E I / DUP 'Z' STO
"Z" →TAG
```

Legt die Funktion der Menütaste 3 fest.

```

CLLCD 1 DISP
1 FREEZE »
« Z » ) )

```

)

Schließt die Liste ab.

MENU

Zeigt das benutzerspezifische Menü an.

»

ENTER **0** EIZ **STO**

Speichert das Programm in *EIZ*.

Beispiel: Nachahmung eines internen Anwendungsprogramms.

Eine Spannungsquelle mit 10 Volt und einem Phasenwinkel 0° speist einen RC-Stromkreis. Es wird ein Strom von 0,37 A mit einem Phasenwinkel 68° gemessen. Wie groß ist die Impedanz der Schaltung?

Wählen Sie das Menü VAR, und starten Sie EIZ.

VAR EIZ

E I Z

Geben Sie den Spannungswert ein.

← **()** 10 **→** **Δ** 0

10.00

E I Z

Speichern Sie den Spannungswert. Geben Sie danach den Stromwert ein und speichern Sie diesen ebenfalls. Bestimmen Sie die Impedanz.

E
← **()** .37 **→** **Δ** 68
I
← Z

:Z: (27,03; -68,00)

4:
3:
2:
1:

E I Z

Welchen Wert muß die komplexe Spannung annehmen, wenn sich die Amplitude des Stroms bei konstanter Impedanz verdoppeln soll?

← **()** .74 **→** **Δ** 68
I
← E

:E: (20,00; -1,36E-10)

4:
3:
2:
1:

E I Z

Übertragen Sie den Wert Z in den Stack zurück.

 Z

|1: (27,03;-68,00)|
|E I 2|

Aufforderung, eine Auswahl zu treffen. Ein benutzerspezifisches Menü kann den Anwender auffordern, während des Programmablaufs eine Entscheidung zu treffen.

Das Programm *WGT* in diesem Abschnitt berechnet das Gewicht eines Gegenstandes entweder in englischen oder in SI-Einheiten. *WGT* baut ein benutzerspezifisches Menü auf, das den Anwender auffordert, das erforderliche Einheitensystem zu wählen. Es folgt die Definitionsliste für das kundenspezifische Menü:

```
{  
  ( "ENGL" « "ENTER Mass  
    in LBM" PROMPT  
    32.2 * » )  
  ( "SI" « "ENTER Mass  
    in KG" PROMPT  
    9.81 * » )  
}
```

Wenn Sie diese Liste in der Variablen *LST* speichern, ist das Programm *WGT* sehr einfach aufgebaut:

Programm:

Kommentar:

«

LST MENU

Zeigt das in *LST* gespeicherte benutzerspezifische Menü an.

»

[ENTER] **[]** WGT **[STO]**

Speichert das Programm in *WGT*.

Das durch *WGT* definierte benutzerspezifische Menü bleibt aktiv, bis Sie ein neues Menü wählen; aus diesem Grund können Sie beliebig viele Berechnungen ausführen.

Beachten Sie, daß das durch *WGT* definierte benutzerspezifische Menü (und das durch *EIZ* definierte Menü) automatisch in der Variablen *CST* gespeichert wird und dabei das vorher gültige benutzerspezifische Menü ersetzt — wenn Sie **[CST]** drücken, nachdem das Programm gelaufen ist, wird das durch *WGT* definierte Menü angezeigt.

Beispiel: Anwendung eines benutzerspezifischen Menüs, um eine Auswahl zu treffen. Lassen Sie *WGT* laufen, um das Gewicht eines Gegenstandes mit der Masse 12.5 kg zu berechnen.

Wählen Sie das Menü **[VAR]**, und starten Sie das Programm.

[VAR] *WGT*

[ENGL] **[SI]** **[]** **[]** **[]** **[]**

Wählen Sie das SI-Einheitensystem.

[SI]

```

ENTER Mass in KG
4:
3:
2:
1:
[ENGL] [SI] [ ] [ ] [ ] [ ]
    
```

Geben Sie die Masse ein, und setzen Sie die Programmausführung fort.

12.5 **[↩]** **[CONT]**

1: 122,63
[ENGL] **[SI]** **[]** **[]** **[]** **[]**

Aufbau eines temporären Menüs

Der Befehl *TMENU* (**[↩]** **[MODES]** **[NXT]** *TMEN*) bewirkt genau dasselbe wie *MENU*, nur ersetzt die Liste nicht den Inhalt von *CST*. Daher bleibt das zuletzt gewählte benutzerspezifische Menü unverändert. Beachten Sie, daß das vorübergehend gültige Menü auch nach Beendigung des Programms aktiv bleibt, bis Sie ein neues Menü wählen. Um das vorherige Menü über das Programm wieder aufzurufen, ist **[]** *MENU* auszuführen.

Das Programm *« LST TMENU »* arbeitet ähnlich wie *WGT*, nur erzeugt es ein temporäres Menü, um den Benutzer zur Wahl eines Einheitensystems aufzufordern.

Befehle, die eine Tastenadresse liefern

Der Befehl WAIT mit dem Argument 0

Wenn Sie 0 als Argument für WAIT verwenden, unterbricht der Befehl den Programmablauf, bis eine gültige Taste gedrückt wird. Er liefert dann eine dreistellige Nummer, aus der die Position der Taste auf der Tastatur hervorgeht, und läßt das Programm wieder weiterlaufen. (Einzelheiten finden Sie im Abschnitt “Umbelegen von Tasten” auf Seite 236 in Kapitel 15.)

(Beachten Sie, daß , , , , oder  selbst keine gültigen Tastenbetätigungen darstellen.)

Der Befehl WAIT mit dem Argument – 1

Der Befehl WAIT mit dem Argument – 1 bewirkt dasselbe wie dieser Befehl mit dem Argument 0, nur wird das aktuelle, zuletzt gewählte Menü ebenfalls angezeigt. Das erlaubt Ihnen, ein Menü zur Bedienung zu erzeugen und anzuzeigen, während das Programm unterbrochen ist. (Beachten Sie, daß ein mit MENU oder TMENU erzeugtes Menü normalerweise erst angezeigt wird, wenn das Programm beendet ist oder mit HALT angehalten wird.)

Der Befehl KEY

Ein Programm kann Sie mit dem Befehl KEY in einer unbestimmten Schleife und durch einen Vergleichstest zu einer einfachen “Ja-Nein”-Entscheidung auffordern. (Unbestimmte Schleifen werden in Kapitel 27 behandelt; Einzelheiten zu Tests finden Sie in Kapitel 26.) KEY liefert solange das Ergebnis “falsch” (0) nach Ebene 1, bis eine Taste gedrückt wird. Sobald eine Taste gedrückt wird, liefert KEY die dreistellige Nummer, aus der die Position der Taste auf der Tastatur hervorgeht, und liefert das Ergebnis “wahr” (1) nach Ebene 1. Wenn Sie zum Beispiel KEY in einer unbestimmten Schleife verwenden und  drücken, liefert KEY 51 nach Ebene 2 und das Ergebnis “wahr” (1) nach Ebene 1.

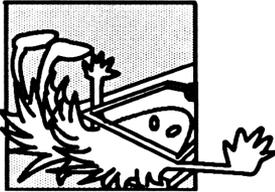
Der folgende Programmausschnitt liefert 1 nach Ebene 1, wenn $\boxed{+}$ gedrückt wird, oder 0 nach Ebene 1 beim Drücken einer beliebigen anderen Taste:

```
◀ ... DO UNTIL KEY END 95 SAME ... ▶
```

Abschalten des HP 48 aus einem Programm

Der Befehl OFF schaltet den HP 48 aus. Geschieht das aus einem Programm heraus, wird das Programm nach dem Wiedereinschalten des Taschenrechners fortgesetzt.

Fehlerbehandlung



Wenn Sie eine unzulässige Operation über die Tastatur vornehmen wollen, wird diese Operation nicht ausgeführt, und es erscheint eine Fehlermeldung. Wollen Sie zum Beispiel einen Vektor und eine reellen Zahl im Stack addieren (+), antwortet der HP 48 mit folgender Meldung:

```
+ Error:
Bad Argument Type
```

und gibt, wenn die Funktion LASTARG aktiviert ist, die Argumente in den Stack zurück. Bei einer unzulässigen Operation in einem Programm geschieht das gleiche, zusätzlich wird das Programm abgebrochen. Betrachten Sie das folgende Programm:

```
« "KEY IN a AND b" "" INPUT OBJ+ + »
```

Wenn Sie dieses Programm verwenden und bei der Aufforderung zur Dateneingabe einen Vektor und eine reelle Zahl eingeben, zeigt das Programm die Meldung `Bad Argument Type` an und bricht die Programmausführung an der Stelle ab, an der der Befehl `+` steht. Um neue Argumente verwenden zu können, müssen Sie das Programm *neu starten*. Bei einem kurzen Programm wie in dem obigen Beispiel stellt diese Art der Fehlerbehandlung kein besonderes Problem dar. Wenn Sie allerdings ein Programm verwenden, das zeitaufwendige Berechnungen

ausführt oder zahlreiche Unterbrechungen für die Zwischeneingabe von Daten aufweist, ist es sehr umständlich, das Programm bei jedem aufgetretenen Fehler wieder erneut von vorne zu starten.

Sie können das Programm so aufbauen, daß der Programmablauf nach dem Auftreten eines Fehlers *fortgesetzt* wird. Das wird durch eine *Fehlerverzweigung* bewirkt. Sie können eine Fehlerverzweigung mit einer der folgenden bedingten Strukturen aufbauen:

- IFERR...THEN...END.
- IFERR...THEN...ELSE...END.

Der Befehl IFERR befindet sich auf Seite 3 des Menüs PRG BRCH.

Die folgenden Befehle stehen für Fehlerverzweigungen zur Verfügung:

Befehle für Fehlerverzweigungen

Tasten	Programmierbarer Befehl	Beschreibung
PRG CTRL (page 3):		
DOERR	DOERR	Verarbeitet einen <i>benutzerdefinierten</i> Fehler. Der Taschenrechner reagiert genau so, als sei ein normaler Fehler aufgetreten — wird der Fehler nicht in einer IFERR-Struktur behandelt, ruft DOERR eine Meldung auf und beendet den Programmablauf.
ERRN	ERRN	Gibt die Fehlernummer des zuletzt aufgetretenen Fehlers als Binärzahl aus. Gibt eine #0 aus, wenn die Fehlernummer durch ERR0 gelöscht wurde.

Befehle für Fehlerverzweigungen (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
<code>ERRM</code>	ERRM	Gibt die Fehlermeldung (eine Zeichenkette) für den zuletzt aufgetretenen Fehler aus. Gibt eine leere Zeichenkette zurück, wenn die Fehlernummer durch ERRO gelöscht wurde.
<code>ERRO</code>	ERRO	Löscht die Fehlernummer des letzten Fehlers, so daß die folgende Ausführung von ERRN eine # <code>0</code> liefert. Löscht außerdem die letzte Fehlermeldung.

Die Struktur IFERR...THEN...END

Die Syntax von IFERR...THEN...END lautet

IFERR *Testausdruck* THEN *Fehleranweisung* END

Tritt ein Fehler während der Auswertung des Testausdrucks auf, wird dieser Fehler ignoriert. Der Rest des Testausdrucks wird nicht mehr ausgewertet, und das Programm springt zur Fehleranweisung. Die Befehle in der Fehleranweisung werden nur dann ausgeführt, wenn ein Fehler während der Auswertung des Testausdrucks auftritt.

Sie können `IFERR` als Schreibhilfe verwenden, um das Gerüst dieser Verzweigung einzugeben:

```
IFERR
THEN
END
```

Beispiel: Eine IFERR...THEN...END-Struktur. Rufen Sie das folgende Programm aus Kapitel 27, Seite 552 auf.

```
« WHILE DUP TYPE 3 == REPEAT  $\Sigma$ + END »
```

Das Programm entnimmt dem Stack eine beliebige Zahl von Vektoren oder Feldern und addiert sie zu einer Statistikmatrix. Das Programm reagiert allerdings mit einer Fehlermeldung, wenn es auf einen Vektor oder ein Feld mit einer anderen Spaltenzahl trifft. Auch wenn sich *nur* Vektoren oder Felder mit derselben Spaltenzahl im Stack befinden, gibt das Programm eine Fehlermeldung aus, nachdem der letzte Vektor dem Stack entnommen wurde.

In der folgenden Version versucht das Programm einfach, das Objekt in Ebene 1 solange zur Statistikmatrix zu addieren, bis ein Fehler auftritt. An dieser Stelle endet es mit der Meldung `DONE`.

Programm:

Kommentar:

```
«
```

```
IFERR
```

Leitet den Testausdruck der Fehlerverzweigung ein.

```
WHILE
```

Beginnt den Testausdruck der inneren Schleife.

```
1
```

1 steht für wahr, deshalb wird die Schleifenanweisung bis zum Auftreten eines Fehlers wiederholt.

```
REPEAT
```

Startet die Schleifenanweisung.

```
 $\Sigma$ +
```

Addiert den Vektor oder das Feld zur Statistikmatrix.

```
END
```

Schließt die innere Schleife ab.

```
THEN
```

Tritt ein Fehler auf bei der Ausführung von Σ + ...

```
"DONE" 1 DISP  
1 FREEZE
```

... erscheint die Meldung `DONE` in den Statuszeilen.

END

Schließt die Fehlerverzweigung ab.

»

Die Struktur IFERR...THEN...ELSE...END

Die Syntax von IFERR...THEN...ELSE...END lautet:

```
IFERR Testausdruck THEN Fehleranweisung ELSE Anweisung END
```

Tritt ein Fehler während der Auswertung des Testausdrucks auf, wird dieser Fehler ignoriert. Der Rest des Testausdrucks wird nicht mehr ausgewertet, und das Programm springt zur Fehleranweisung. Wenn kein Fehler auftritt, setzt das Programm nach Beendigung des Testausdrucks den normalen Ablauf fort.

Sie können  IFERR als Schreibhilfe verwenden, um das Gerüst der Verzweigung einzugeben:

```
IFERR  
THEN  
ELSE  
END
```

Beispiel: Eine IFERR...THEN...ELSE...END-Struktur. Das folgende Programm fordert Sie auf, zwei Zahlen einzugeben und addiert sie anschließend. Wenn Sie nur eine Zahl eingeben, reagiert das Programm mit einer Fehlermeldung in der Anzeige und fordert erneut eine Eingabe.

Programm:

Kommentar:

»

```
DO
```

Beginnt mit der äußeren Schleife.

```
"KEY IN a AND b" " "  
INPUT OBJ→
```

Fordert die Eingabe von zwei
Zahlen.

UNTIL	Beginnt den Testausdruck.
IFERR	Beginnt die Fehlerverzweigung
+	Addiert den Inhalt der Ebenen 1 und 2.
THEN	Tritt ein Fehler auf, ...
ERRM 5 DISP	... so wird ERRM ausgeführt, um
2 WAIT	zwei Sekunden lang die
Ø	Fehlermeldung Too Few Arguments (Zu wenige Argumente) anzuzeigen, und anschließend wird eine Ø (falsch) für den Befehl END der äußeren Schleife im Stack abgelegt.
ELSE	Tritt kein Fehler auf, ...
1	... so wird eine 1 (wahr) für den Befehl END der äußeren Schleife im Stack abgelegt.
END	Schließt die Fehlerverzweigung ab.
END	Schließt die äußere Schleife ab. Steht nach Ausführung der Fehlerverzweigung eine Ø im Stack, springt das Programm mit diesem END an die Stelle zurück, an der der Bediener zur Eingabe der Zahlen aufgefordert wird. Im anderen Fall beendet der Befehl das Programm.

❖

Benutzerdefinierte Fehler

Es ist möglich, daß Sie in einem Programm an einer Stelle einen Fehler erzeugen wollen, an der normalerweise kein Fehler auftritt. Es ist zum Beispiel denkbar, daß Sie eine Fehlermeldung erzeugen wollen, wenn die Summe von zwei Zahlen auf dem Stack größer als 10 ist. Das können Sie mit dem Befehl DOERR erreichen. DOERR veranlaßt ein Programm dazu, sich genauso zu verhalten, als sei ein normaler Fehler im Programmablauf aufgetreten. Der DOERR-Fehler kann in einer Fehlerverzweigung (IFERR-Struktur) behandelt werden. Geschieht dies nicht, so wird die Ausführung des Programms an der Stelle beendet, an der der Befehl DOERR im Programm steht. DOERR verwendet *eines* der folgenden Argumente vom Stack:

- Eine Zeichenkette, die als Meldung aufgefaßt wird. (ERRM gibt diese Zeichenkette aus und ERRN liefert #70000h.)
- Eine reelle Zahl oder eine Binärzahl. In diesem Fall wird die entsprechende interne Fehlermeldung angezeigt. (ERRM und ERRN geben die entsprechende Fehlermeldung beziehungsweise Fehlernummer aus.) □ DOERR ist gleichbedeutend mit [ATTN]; das heißt, die Programmausführung wird abgebrochen, und es wird keine Meldung angezeigt. (In diesem Fall bleiben die Fehlermeldung bzw. Fehlernummer, die ERRM bzw. ERRN ausgeben, unverändert.)

Das folgende Programm bricht die Ausführung ab, wenn drei Objekte in der Liste der Ebene 1 enthalten sind.

```
⋄  
OBJ→  
IF 3 SAME  
THEN "3 OBJECTS IN LIST" DOERR  
END  
⋄
```

In diesem Programm beendet DOERR die Ausführung des Programms. Als Alternative können Sie DOERR im Testausdruck einer Fehlerverzweigung verwenden, um die Fortsetzung des Programms freizugeben.

Weitere Programmierbeispiele



Die Programme in diesem Kapitel sollen einige Programmierkonzepte verdeutlichen, die in den vorhergehenden Kapiteln bereits besprochen wurden. Außerdem werden einige neue Ideen vorgestellt. Dieses Kapitel bietet Ihnen also die Gelegenheit, Ihre Programmierfähigkeiten zu verbessern, und es werden Ihnen am Ende zusätzliche Funktionen für Ihren Taschenrechner zur Verfügung stehen.

Am Ende jedes Programms wird die *Größe* des Programms in Bytes sowie die *Prüfsumme* angegeben. Die Prüfsumme ist eine Binärzahl, die das Programm eindeutig kennzeichnet und aus seinem Inhalt abgeleitet wird. Sie können mit Hilfe des Befehls BYTES (◀ **MEMORY** BYTES) überprüfen, ob Sie das Programm fehlerfrei eingegeben haben. Er benötigt den Programmnamen in Ebene 1 und schreibt die Prüfsumme des Programms nach Ebene 2 sowie seine Größe in Bytes nach Ebene 1.

Fibonacci-Zahlen

Dieser Abschnitt enthält drei Programme — zwei von ihnen demonstrieren zwei unterschiedliche Methoden zur Lösung des folgenden Problems.

Gegeben sei eine Zahl n ; berechnen Sie die n te Fibonacci-Zahl F_n , wobei

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

- *FIB1* ist eine *rekursive* anwenderdefinierte Funktion — ihre definierende Prozedur enthält ihren eigenen Namen. *FIB1* ist kurz.
- *FIB2* ist eine anwenderdefinierte Funktion mit einer bestimmten Schleife. Sie ist umfangreicher und komplizierter aufgebaut als *FIB1*, aber sie ist schneller.

Das dritte Programm *FIBT* ruft die beiden Programme *FIB1* und *FIB2* auf und berechnet die Ausführungszeit für jedes Unterprogramm.

FIB1 (Fibonacci-Zahlen, Rekursive Version)

Argumente	Ergebnisse
1: n	1: F_n

Verfahren

- IFTE (IF-THEN-ELSE-Funktion). Die definierende Prozedur für *FIB1* enthält die bedingte Funktion IFTE, die ihr Argument entweder vom Stack holen oder in algebraischer Syntax verarbeiten kann. (*FIB2* verwendet die Verzweigungsstruktur IF ... THEN ... ELSE ... END.)
- Rekursion. Die definierende Prozedur für *FIB1* wird als Ausdruck von *FIB1* geschrieben, genau so wie F_n als Ausdruck von F_{n-1} und F_{n-2} definiert ist.

Programm:**Kommentar:**

⌘

→ n

Definiert die lokale Variable n .

'

Leitet die definierende Prozedur ein, einen algebraischen Ausdruck.

IFTE($n \leq 1$,Wenn $n \leq 1 \dots$

n,

... dann $F_n = n \dots$ FIB1($n-1$)+FIB1($n-2$))... sonst $F_n = F_{n-1} + F_{n-2}$.

'

Schließt die definierende Prozedur ab.

⌘

[ENTER] [] FIB1 [STO]

Speichert das Programm in *FIB1*.**Prüfsumme:** # 41467d**Bytes:** 113,5**Beispiel.** Berechnen Sie F_6 . Berechnen Sie F_{10} unter Verwendung algebraischer Syntax.Berechnen Sie zuerst F_6 .[VAR]
6 FIB11: 8
[EQ] [FIB1] [FIB2] [FIBT] [PMD] [PRESE]Berechnen Sie als nächstes F_{10} und verwenden Sie dabei algebraische Syntax.

[] FIB1 [←] [()] 10 [EVAL]

2: 8
1: 55
[EQ] [FIB1] [FIB2] [FIBT] [PMD] [PRESE]

FIB2 (Fibonacci-Zahlen, Schleifenversion)

Argumente	Ergebnisse
1: n	1: F_n

Verfahren

- IF...THEN...ELSE...END. *FIB2* verwendet die Programmstruktur der bedingten Verzweigung. (*FIB1* verwendet IFTE.)
- START...NEXT (Bestimmte Schleife). Um F_n zu berechnen, beginnt *FIB2* mit F_0 und F_1 und durchläuft mehrfach eine Schleife, um aufeinanderfolgende Werte F_i zu berechnen.

Programm:

Kommentar:

⌘

→ n

Erzeugt eine lokale Variable.

⌘

Leitet die definierende Prozedur ein, ein Programm.

IF $n \leq 1$

Wenn $n \leq 1$...

THEN n

... dann $F_n = n$.

ELSE

Leitet die ELSE-Anweisung ein.

Ø 1

Holt F_0 und F_1 in den Stack.

2 n

Von 2 bis n ...

START

... wird die folgende Schleife ausgeführt:

DUP

Dupliziert das neueste F (beim ersten Durchlauf F_1).

ROT

Verschiebt den vorhergehenden Wert von F (beim ersten Durchlauf F_0) nach Ebene 1.

+	Berechnet den nächsten Wert F (beim ersten Durchlauf F_2).
NEXT	Führt den nächsten Schleifendurchlauf aus.
SWAP DROP	Entfernt F_{n-1} aus dem Stack.
END	Schließt die ELSE-Anweisung ab.
⌘	Schließt die definierende Prozedur ab.
⌘	Beendet das Programm.
[ENTER] [] FIB2 [STO]	Speichert das Programm in FIB2.

Prüfsumme: # 51820d
Bytes: 89

Beispiel. Berechnen Sie F_6 und F_{10} . Beachten Sie, daß *FIB2* schneller ist als *FIB1*.

Berechnen Sie F_6 .

[VAR]
6 **[FIB2]**

1:	8
[BOISP] [FIB1] [FIB2] [FIBT] [PND] [PRESE]	

Berechnen Sie F_{10} .

10 **[FIB2]**

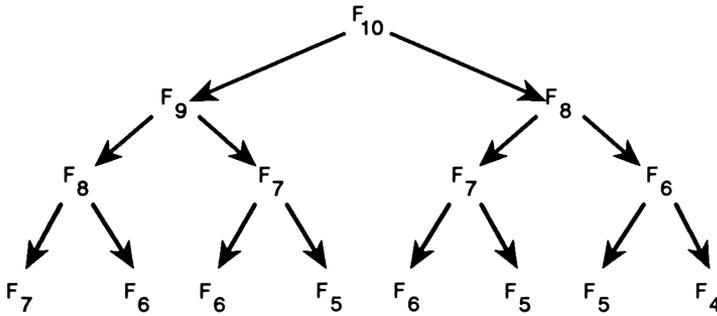
2:	8
1:	55
[BOISP] [FIB1] [FIB2] [FIBT] [PND] [PRESE]	

FIBT (Vergleich der Programmausführungszeit)

FIB1 berechnet die Zwischenwerte F_i mehrfach, während *FIB2* jeden Zwischenwert F_i nur einmal bestimmt. Daraus folgt, daß *FIB2* schneller ist. Der Unterschied in der Geschwindigkeit wird mit wachsendem n größer, weil die erforderliche Zeit zur Ausführung von *FIB1* exponentiell mit n zunimmt, während die erforderliche Ausführungszeit für *FIB2* nur linear mit n wächst.

Das folgende Diagramm zeigt die ersten Schritte von *FIB1* zur Berechnung von F_{10} . Beachten Sie die Anzahl der Zwischenschritte bei

der Berechnung: 1 in der ersten Reihe, 2 in der zweiten Reihe, 4 in der dritten Reihe und 8 in der vierten Reihe.



FIBT benutzt den Befehl *TICKS*, um die Ausführungszeit von *FIB1* und *FIB2* für einen gegebenen Wert n zu ermitteln.

Argumente	Ergebnisse
1: n	z : F_n z : Ausführungszeit <i>FIB1</i> : z : Ausführungszeit <i>FIB2</i> :

Verfahren

- Strukturierte Programmierung. *FIBT* ruft beide Programme *FIB1* und *FIB2* auf.
- Verwendung der Uhr des Taschenrechners durch das Programm. *FIBT* benutzt den Befehl *TICKS*, um den Zeitpunkt von Anfang und Ende jedes Unterprogramms zu ermitteln.
- Dialogprogrammierung. *FIBT* versieht jede Ausführungszeit mit einer erläuternden Meldung.

Programm:*
*

```
DUP TICKS SWAP FIB1
SWAP TICKS SWAP
```

```
- B→R 8192 /
```

```
"FIB1 TIME"
→TAG
```

```
ROT TICKS SWAP FIB2
TICKS
```

```
SWAP DROP SWAP
- B→R 8192 /
```

```
"FIB2 TIME"
→TAG
```

*
*

```
[ENTER] [] FIBT [STO]
```

Prüfsumme: # 22248d
Bytes: 135

Kommentar:

Dupliziert n , führt danach *FIB1* aus und legt Anfangs- und Endzeit im Stack ab.

Berechnet die abgelaufene Zeit, wandelt sie in eine reelle Zahl um und rechnet diese Zahl in Sekunden um. Das mit *FIB1* bestimmte Ergebnis steht danach in Ebene 2.

Kennzeichnet die Ausführungszeit.

Führt *FIB2* aus und ermittelt Anfangs- und Endzeit.

Entfernt das von *FIB2* gelieferte Ergebnis aus dem Stack (*FIB1* hat das gleiche Ergebnis geliefert). Berechnet die Ausführungszeit von *FIB2* und wandelt das Ergebnis in Sekunden um.

Kennzeichnet die Ausführungszeit.

Speichert das Programm in *FIBT*.

Beispiel. Berechnen Sie F_{13} und vergleichen Sie die Ausführungszeiten, die mit beiden Methoden ermittelt wurden.

Wählen Sie das Menü VAR und führen Sie die Berechnung aus.

VAR

13 FIBT

```
{ HOME }
3: 233
2: FIB1 TIME: 33.8876...
1: FIB2 TIME:
   .127075195312
FIB1 FIB2 FIBT
```

F_{13} ist 233. *FIB2* benötigt 0,13 Sekunden für die Ausführung. *FIB1* benötigt 33,9 Sekunden. (Ihre Ergebnisse können je nach Speicherinhalt Ihres Taschenrechners von diesen Zahlen etwas abweichen.)

Anzeigen einer Binärzahl

In diesem Abschnitt werden drei Programme beschrieben:

- *PAD* ist ein Dienstprogramm, das ein Objekt in eine Zeichenkette umwandelt und für eine rechtsbündige Anzeige vorbereitet.
- *PRESERVE* ist ein Dienstprogramm für die Anwendung in Programmen, die den Status des Taschenrechners ändern (Winkelmodus, Zahlenbasis für Binärzahlen, usw.).
- *BDISP* stellt eine Binärzahl als Hexadezimal-, Dezimal-, Oktal- und Dualzahl dar. Es ruft *PAD* auf, um die dargestellten Zahlen rechtsbündig anzuzeigen, und es ruft *PRESERVE* auf, um die binäre Zahlenbasis wieder in den ursprünglichen Zustand zu versetzen.

PAD (Auffüllen mit führenden Leerstellen)

PAD wandelt ein Objekt in eine Zeichenkette um und fügt am Anfang Leerstellen ein, wenn die Zeichenkette weniger als 23 Zeichen enthält.

Wird eine kurze Zeichenkette mit *DISP* dargestellt, erscheint sie *linksbündig*: ihr erstes Zeichen erscheint am linken Rand des Anzeigefeldes. Die Position des letzten Zeichens hängt von der Länge der Zeichenkette ab. Durch das Einfügen von Leerstellen am Anfang einer kurzen Zeichenkette verschiebt *PAD* die Position des letzten Zeichens nach rechts. Ist die Zeichenkette (einschließlich der führenden Leerstellen) 23 Zeichen lang, erscheint sie *rechtsbündig*; ihr letztes Zeichen steht am rechten Rand des Anzeigefeldes. *PAD* hat keinerlei Wirkung auf Zeichenketten mit mehr als 22 Zeichen.

Argumente	Ergebnisse
1: <i>Objekt</i>	1: " <i>Objekt</i> "

Verfahren

- **WHILE ... REPEAT ... END** (unbestimmte Schleife). Die WHILE-Anweisung enthält einen Test, mit dem bestimmt wird, ob die REPEAT-Anweisung und der Test ein weiteres Mal auszuführen sind (Testergebnis "wahr") oder ob die REPEAT-Anweisung zu überspringen und das Unterprogramm zu verlassen ist (Testergebnis "falsch").
- **Operationen mit Zeichenketten.** *PAD* demonstriert, wie ein Objekt in eine Zeichenkette umgewandelt wird, zählt die Anzahl der Zeichen und verknüpft zwei Zeichenketten.

Programm:

Kommentar:

⌘

→STR

Stellt sicher, daß das Objekt eine Zeichenkette ist. (Zeichenketten werden von diesem Befehl nicht beeinflußt.)

WHILE

Leitet die WHILE-Anweisung ein.

DUP SIZE 22 <

Enthält die Zeichenkette weniger als 23 Zeichen ?

REPEAT

Leitet die REPEAT-Anweisung ein.

" " SWAP +

Fügt eine führende Leerstelle ein.

END

Schließt die REPEAT-Anweisung ab.

⌘

ENTER **1** **PAD** **STO**

Speichert das Programm in *PAD*.

Prüfsumme: # 38912d
Bytes: 61,5

Die Anwendung von *PAD* wird im Programm *BDISP* demonstriert.

PRESERVE (Sichern und Wiederherstellen des vorherigen Zustands)

Wenn ein Programm im Stack steht, speichert *PRESERVE* den aktuellen Taschenrechner-(Flag-)Status, führt das Programm aus und stellt dann den vorherigen Zustand wieder her.

Arguments	Results
1: « Programm »	1: (Programmergebnis)
1: 'Name des Programms '	1: (Programmergebnis)

Verfahren

- **RCLF und STOF.** *PRESERVE* verwendet *RCLF* (*Flags aufrufen*), um den aktuellen Status des Taschenrechners in Form einer Binärzahl in den Stack zu holen und *STOF* (*Flags speichern*), um den Status später durch Verwendung dieser Binärzahl wiederherzustellen.
- **Struktur lokaler Variablen.** *PRESERVE* erzeugt eine Struktur lokaler Variablen, die die Binärzahl vorübergehend aus dem Stack entfernt; ihre definierende Prozedur führt das Programm aus, holt die Binärzahl anschließend in den Stack zurück und führt *STOF* aus.

Programm:

⌘

RCLF

Kommentar:

Legt eine Liste von zwei 64-Bit-Binärzahlen im Stack ab, die den Zustand der 64 Systemflags und der 64 Benutzerflags wiedergeben.

→ f	Speichert die Liste in der lokalen Variablen <i>f</i> .
⋈	Leitet die definierende Prozedur ein.
EVAL	Führt das Programm aus, das als Argument in Ebene 1 des Stack liegt.
f STOF	Holt Liste zurück in den Stack und stellt anschließend den Zustand aller Flags wieder her.
⋈	Beendet die definierende Prozedur.
⋈	
[ENTER] [] PRESERVE [STO]	Speichert das Programm in <i>PRESERVE</i> .

Prüfsumme: # 21528d
Bytes: 46,5

Die Anwendung von *PRESERVE* wird im Programm *BDISP* demonstriert.

BDISP (Binäranzeige)

BDISP stellt eine (reelle oder binäre) Zahl als Hexadezimal-, Dezimal-, Oktal- und Dualzahl dar.

Argumente	Ergebnisse
1: # <i>n</i>	1: # <i>n</i>
1: <i>n</i>	1: <i>n</i>

Verfahren

- IFERR ... THEN ... END (Fehlerverzweigung). *BDISP* enthält den Befehl R→B (*reell in binär*), um auch reelle Zahlen als Argumente verarbeiten zu können. Dieser Befehl führt allerdings zu einer Fehlermeldung, wenn das Argument *bereits* eine Binärzahl ist. Damit das Programm bei Auftreten eines Fehlers nicht abbricht, wird der Befehl R→B in eine IFERR-Anweisung eingeschlossen. Tritt ein

Fehler auf, ist keine Maßnahme erforderlich (da eine Binärzahl ein gültiges Argument ist); aus diesem Grund enthält die THEN-Anweisung keine Befehle.

- LASTARG aktivieren. Bei Auftreten eines Fehlers wird LASTARG benötigt, um das Argument (die Binärzahl) in den Stack zurückzuholen. *BDISP* setzt das Flag -55 zurück, um die Wiederherstellung mit LASTARG zu ermöglichen.
- FOR ... NEXT loop (bestimmte Schleife mit Zähler). *BDISP* durchläuft eine Schleife von 1 bis 4, wobei jedesmal (die Zahl) n mit unterschiedlicher Zahlenbasis in einer anderen Zeile angezeigt wird. Der Schleifenzähler (in diesem Programm mit j bezeichnet) ist eine lokale Variable. Sie wird vom Programmteil FOR ... NEXT (hier also nicht durch den Befehl \rightarrow) erzeugt und wird automatisch von NEXT erhöht.
- Programme ohne Namen als Argumente. Ein Programm, das nur durch seine Begrenzungszeichen « und » gekennzeichnet ist (und nicht in einer Variablen gespeichert ist), wird nicht automatisch ausgeführt; es wird einfach in den Stack gebracht und kann als Argument für ein Unterprogramm verwendet werden. *BDISP* zeigt zwei Anwendungsmöglichkeiten für Programme ohne Namen als Argumente.
 1. *BDISP* enthält ein Hauptprogramm als Argument für *PRESERVE* und einen Aufruf von *PRESERVE*. Dieses Argument wird also in den Stack übertragen und durch *PRESERVE* ausgeführt.
 2. Es gibt vier weitere Programme, die als Argumente dienen und die die Wirkungsweise der Schleife festlegen. Jedes dieser Programme enthält einen Befehl zur Änderung der Zahlenbasis, und bei jeder Wiederholung der Schleife wird eines dieser Argumente ausgewertet.

Wenn *BDISP* eine lokale Variable für n erzeugt, ist die definierende Prozedur ein unbenanntes Programm. Da dieses Programm eine definierende Prozedur für eine Struktur lokaler Variablen ist, wird es automatisch ausgeführt.

Erforderliche Programme

- *PAD* (Seite 601) erweitert eine Zeichenkette auf 23 Zeichen, so daß sie von DISP rechtsbündig dargestellt wird.
- *PRESERVE* (Seite 602) speichert den aktuellen Status, führt das eingebettete Hauptprogramm aus und stellt den Status wieder her.

Programm:**Kommentar:**

«

«

Leitet das eingebettete Hauptprogramm ein.

DUP

Dupliziert n .

-55 CF

Setzt das Flag -55 zurück, um LASTARG zu aktivieren.

IFERR

Leitet Fehlerverzweigung ein.

R→B

Wandelt n in eine Binärzahl um.

THEN

Falls ein Fehler auftrat. . .

END

. . . erfolgt keine Reaktion (die THEN-Anweisung enthält keine Befehle).

→ n

Erzeugt eine lokale Variable n .

«

Beginnt die definierende Prozedur für die Struktur lokaler Variablen.

CLLCD

Löscht das Anzeigefeld.

« BIN »

Erzeugt das eingebettete Programm für BIN.

« OCT »

Erzeugt das eingebettete Programm für OCT.

« DEC »

Erzeugt das eingebettete Programm für DEC.

« HEX »

Erzeugt das eingebettete Programm für HEX.

1 4

Anfangs- und Endwert des Zählers.

FOR j

Beginnt den Schleifendurchlauf mit dem Zählerwert j .

EVAL	Führt eines der Programme zur Einstellung der Zahlenbasis aus (beim ersten Durchlauf das Programm für HEX).
$n \rightarrow \text{STR}$	Bildet eine Zeichenkette, die n auf der aktuellen Zahlenbasis darstellt.
PAD	Füllt die Zeichenkette auf 23 Zeichen auf.
$j \text{ DISP}$	Stellt die Zeichenkette in der j ten Zeile dar.
NEXT	Erhöht j und wiederholt den Schleifendurchlauf.
»	Schließt die definierende Prozedur ab.
$\exists \text{ FREEZE}$	Friert den Status und die Stackbereiche ein.
»	Beendet das eingebettete Hauptprogramm.
PRESERVE	Speichert den aktuellen Status, führt das Hauptprogramm aus und stellt den Status wieder her.
»	
[ENTER] [] BDISP [STO]	Speichert das Programm in <i>BDISP</i> .

Prüfsumme: # 18055d
Bytes: 191

Beispiel. Schalten Sie auf die Zahlenbasis DEC für die dezimale Darstellung, stellen Sie # 100 in allen vier Zahlensystemen dar, und kontrollieren Sie, ob *BDISP* die Zahlenbasis DEC wiederhergestellt hat.

Löschen Sie den Stack und wählen Sie das Menü MTH BASE. Stellen Sie sicher, daß die Zahlenbasis DEC ist geben Sie # 100 ein.

```

[→] [CLR]
[MTH] BASE
DEC
[→] [#] 100 [ENTER]
    
```

```

1: # 100d
HEX DEC OCT BIN STMS ROWS
    
```

Starten Sie *BDISP*.

```
[VAR] BDISP
```

```

# 64h
# 100d
# 144o
# 1100100b
BDISP FIE1 FIE2 FIE3 PWD PRESB
    
```

Kehren Sie zur normalen Stackanzeige zurück, und kontrollieren Sie die aktuelle Basis.

```

[ATTN]
[MTH] BASE
    
```

```
HEX DEC OCT BIN STMS ROWS
```

Obwohl das eingebettete Hauptprogramm zuletzt die Basis BIN verwendete, hat *PRESERVE* die Basis DEC wiederhergestellt.

Verwenden Sie die Zahl 144, um zu kontrollieren, ob *BDISP* auch mit reellen Zahlen arbeitet.

```

[VAR]
144 BDISP
    
```

```

# 90h
# 144d
# 220o
# 10010000b
BDISP FIE1 FIE2 FIE3 PWD PRESB
    
```

Medianwert von statistischen Daten

Dieser Abschnitt enthält drei Programme:

- *SORT* ordnet die Elemente einer Liste.
- *LMED* berechnet den Medianwert der geordneten Liste.
- *MEDIAN* verwendet *SORT* und *LMED*, um den Medianwert der aktuellen statistischen Daten zu berechnen.

SORT (Ordnen einer Liste)

SORT sortiert eine Liste von reellen Zahlen in aufsteigender Ordnung.

Argumente	Ergebnisse
1: \langle Liste \rangle	1: \langle Geordnete Liste \rangle

Verfahren

- Sortierverfahren “Bubble Sort”. *SORT* vergleicht benachbarte Zahlen miteinander und schiebt die größere Zahl nach hinten; das Programm beginnt mit der ersten und der zweiten Zahl in der Liste. Dieser Vorgang wird einmal die ganze Liste hindurch ausgeführt, um die größte Zahl an die letzte Stelle in der Liste zu setzen; danach ein weiteres Mal, um die zweitgrößte Zahl an die vorletzte Position zu setzen, usw..
- Bestimmte Schleifen mit Schachtelung. Die äußere Schleife legt bei jedem ihrer Durchläufe die Endposition der inneren Schleife fest; die innere Schleife läuft von 1 bis zur Endposition; der Schleifenzähler wird dabei nach jedem Durchlauf erhöht.
- Verschachtelte Strukturen lokaler Variablen. *SORT* enthält zwei Strukturen lokaler Variablen; die zweite ist in die definierende Prozedur (ein Programm) der ersten Struktur eingebettet. Diese Verschachtelung macht den Programmaufbau komfortabler: es ist einfacher, die erste lokale Variable zu erzeugen, sobald ihr Wert berechnet ist und dabei ihren Wert aus dem Stack herauszunehmen, anstatt beide Werte zu berechnen und beide lokale Variablen auf einmal zu erzeugen.
- FOR ... STEP und FOR ... NEXT (bestimmte Schleifen). *SORT* verwendet zwei Zähler: - 1 STEP verringert den Zähler der äußeren Schleife nach jedem Durchlauf; NEXT erhöht den Zähler der inneren Schleife nach jedem Schritt um 1.

Programm:

⌘

DUP SIZE 1 - 1

FOR j

1 j

FOR k

k GETI → n1

⌘

GETI → n2

⌘

DROP

IF n1 n2 >

THEN

k n2 PUTI

Kommentar:

Von der vorletzten Stelle zur ersten Stelle ...

... startet die äußere Schleife mit dem Zähler j .

Von der ersten Stelle bis zur j ten Stelle ...

... startet die innere Schleife mit dem Zähler k .

Nimmt die k te Zahl in der Liste und speichert sie in einer lokalen Variablen n_1 .

Leitet die definierende Prozedur (ein Programm) für die äußere Struktur lokaler Variablen ein.

Nimmt die nächste Zahl in der Liste und speichert sie in einer lokalen Variablen n_2 .

Leitet die definierende Prozedur (ein Programm) für die innere Struktur lokaler Variablen ein.

Entfernt den Index, der von GETI erzeugt wurde, aus dem Stack.

Wenn die beiden Zahlen in der falschen Reihenfolge stehen ...

... dann geschieht folgendes:

... die zweite Zahl (die vorher an der $k + 1$ ten Stelle stand) wird an die k te Stelle geschrieben;

n1 PUT	... die k te wird an die $k + 1$ te Stelle geschrieben.
END	Beendet die THEN-Anweisung.
⌘	Schließt die definierende Prozedur für die innere Struktur ab.
⌘	Schließt die definierende Prozedur für die äußere Struktur ab.
NEXT	Erhöht k und durchläuft erneut die innere Schleife.
-1 STEP	Verringert j und durchläuft erneut die äußere Schleife.
⌘	
[ENTER] [] SORT [STO]	Speichert das Programm in <i>SORT</i> .

Prüfsumme: # 15011d
Bytes: 144

Beispiel. Sortieren Sie die Liste { 8 3 1 2 5 }.

Wählen Sie das Menü VAR, geben Sie die Liste ein und starten Sie *SORT*.

```

[VAR]
[←] [{}] 8 [SPC] 3 [SPC]
1 [SPC] 2 [SPC] 5 [ENTER]
SORT

```

```

1: { 1 2 3 5 8 }
[EDIT] [LMED] [MEDM] [OOLN] [MULTI] [ERCD]

```

LMED (Medianwert einer Liste)

Gegeben sei eine geordnete Liste, dann liefert *LMED* den Medianwert. Enthält die Liste eine ungerade Zahl von Elementen, dann ist der Medianwert der Wert des Elementes in der Mitte. Enthält die Liste eine gerade Anzahl von Elementen, dann ist der Medianwert der arithmetische Mittelwert aus den beiden Elementen direkt unterhalb und direkt oberhalb der Mitte der Liste.

Argumente	Ergebnisse
1: { <i>Geordnete Liste</i> }	1: <i>Medianwert der geordneten Liste</i>

Verfahren

- FLOOR und CEIL. Bei einer ganzen Zahl liefern FLOOR und CEIL beide diese ganze Zahl; ist eine gebrochene Zahl gegeben, liefern FLOOR und CEIL die aufeinanderfolgenden ganzen Zahlen, von denen die gebrochene Zahl eingeschlossen wird.

Programm:

Kommentar:

⌘

DUP SIZE

Dupliziert die Liste und ermittelt anschließend ihre Länge.

1 + 2 ✓

Berechnet die Position der Mitte der Liste (gebrochene Zahl bei Listen mit einer geraden Anzahl von Elementen)

→p

Speichert die Mittenposition in der lokalen Variablen *p*.

⌘

Leitet die definierende Prozedur (ein Programm) für die Struktur lokaler Variablen ein.

DUP

Dupliziert die Liste.

p FLOOR GET

Nimmt die Zahl bei oder direkt unterhalb der mittleren Position.

SWAP

Überträgt die Liste nach Ebene 1.

p CEIL GET

Nimmt die Zahl bei oder direkt oberhalb der mittleren Position.

+ 2 ✓

Berechnet den arithmetischen Mittelwert der beiden Zahlen in der mittleren Position oder in den benachbarten Positionen.

»

Schließt die definierende Prozedur ab.

»

[ENTER] **[]** LMED **[STO]**

Speichert das Programm in *LMED*.

Prüfsumme: # 3682d

Bytes: 77

Beispiel. Berechnen Sie den Medianwert der Liste, die Sie mit *SORT* geordnet haben.

Legen Sie die Liste im Stack ab, falls erforderlich, wählen Sie das Menü *VAR* und starten Sie *LMED*.

[←] **[{]** 1 2 3 5 8 **[ENTER]**

[VAR] LMED

[1:] **[3]**
[SORT] **[LMED]** **[MEDIAN]** **[DOWN]** **[MULTI]** **[EXEC]**

MEDIAN (Medianwert von statistischen Daten)

MEDIAN liefert einen Vektor, der die Medianwerte der Spalten der statistischen Datenmatrix enthält.

Argumente	Ergebnisse
1:	1: $[x_1 \ x_2 \ \dots \ x_m]$

Verfahren

- Felder, Listen und Stackelemente. *MEDIAN* wählt aus *SDAT* eine Datenspalte in Vektorform. *MEDIAN* zerlegt diesen Vektor in seine Elemente, legt sie im Stack ab und faßt sie dann in einer Liste zusammen. Aus dieser Liste wird der Medianwert mit *SORT* und *LMED* berechnet.

Der Medianwert für die *m*te Spalte wird zuerst berechnet, und die Berechnung des Medianwertes für die erste Spalte erfolgt zuletzt. Jeder Medianwert wird nach seiner Berechnung in eine Stackebene oberhalb der vorher berechneten Medianwerte übertragen. Nachdem alle Medianwerte berechnet wurden und in der richtigen Reihenfolge

im Stack stehen, werden sie zu einem Vektor zusammengefaßt.

- **FOR ... NEXT** (bestimmte Schleife mit Zähler). *MEDIAN* verwendet eine Schleife, um den Medianwert jeder einzelnen Schleife zu berechnen. Da die Berechnung der Medianwerte in umgekehrter Reihenfolge durchgeführt wird (die letzte Spalte zuerst), verwendet das Programm den Zähler dazu, die Reihenfolge der Medianwerte umzukehren.

Erforderliche Programme

- *SORT* (Seite 609) sortiert eine Liste in aufsteigender Ordnung.
- *LMED* (Seite 611) berechnet den Medianwert der geordneten Liste.

Programm:

Kommentar:

⌘

RCLΣ

Erzeugt zur Sicherheit eine Kopie der aktuellen Statistikmatrix ΣDAT im Stack.

DUP SIZE

Legt die Liste $\{ n m \}$ im Stack ab, wobei n die Anzahl der Zeilen in ΣDAT und m die Anzahl der Spalten darstellt.

OBJ→ DROP

Zerlegt die Liste, legt n und m im Stack ab und entfernt die Länge der Liste vom Stack.

→ n m

Erzeugt lokale Variablen für n und m .

⌘

Leitet die definierende Prozedur (ein Programm) für die Struktur lokaler Variablen ein.

'ΣDAT' TRN

Transponiert ΣDAT . Jetzt ist n die Anzahl der Spalten in ΣDAT und m die Anzahl der Zeilen. (Geben Sie das Zeichen Σ ein, indem Sie   drücken, und löschen Sie anschließend die Klammern.)

1 m	Bezeichnet die erste und die letzte Zeile.
FOR j	Führt in jeder Zeile folgende Operation aus:
$\Sigma-$	Wählt die letzte Zeile in ΣDAT . Beim ersten Durchlauf ist dies die m te Zeile, die der m ten Spalte in der Ausgangsmatrix ΣDAT entspricht. (Geben Sie den Befehl $\Sigma-$ ein, indem Sie  [STAT]  $\Sigma+$ drücken.)
OBJ \rightarrow DROP	Legt die Zeilenelemente im Stack ab. Entfernt die Indexliste $\{ n \}$ vom Stack, da n bereits in einer lokalen Variablen gespeichert ist.
$n \rightarrow$ LIST	Stellt eine Liste aus n Elementen zusammen.
SORT	Sortiert die Liste.
LMED	Berechnet den Medianwert der Liste.
j ROLLD	Verschiebt den Medianwert in die richtige Ebene des Stacks.
NEXT	Erhöht j und durchläuft die Schleife erneut.
$m \rightarrow$ ARRAY	Faßt alle Medianwerte zu einem Vektor mit m Elementen zusammen.
»	Schließt die definierende Prozedur ab.
SWAP	Überträgt die Ausgangsmatrix ΣDAT auf Ebene 1.
STO Σ	Stellt den ursprünglichen Wert von ΣDAT wieder her.

✳

ENTER **▮** **MEDIAN** **STO**

Speichert das Programm in *MEDIAN*.

Prüfsumme: # 19502d
Bytes: 129,5

Beispiel. Berechnen Sie den Medianwert der folgenden Daten.

$$\begin{bmatrix} 18 & 12 \\ 4 & 7 \\ 3 & 2 \\ 11 & 1 \\ 31 & 48 \\ 20 & 17 \end{bmatrix}$$

Diese Matrix enthält zwei Datenspalten, so daß *MEDIAN* einen Vektor mit zwei Elementen liefert.

Geben Sie die Matrix ein.

➡ **MATRIX**
 18 **ENTER** 12 **ENTER** **▼**
 4 **ENTER** 7 **ENTER**
 3 **ENTER** 2 **ENTER**
 11 **ENTER** 1 **ENTER**
 31 **ENTER** 48 **ENTER**
 20 **ENTER** 17 **ENTER**
ENTER

```
1: [[ 18 12 ]
    [ 4 7 ]
    [ 3 2 ]
    [ 11 1 ]
    ]
SORT LMED MEDIAN DOWN MULTI ESCD
```

Speichern Sie die Matrix in ΣDAT .

↶ **STAT** **Σ+**

```
ΣDAT(6)=[ 20 17 ]
ΣDAT(7)=
4:
3:
2:
1:
Σ+ CLΣ NEW EDITΣ STOΣ CAT
```

Berechnen Sie den Medianwert.

[VAR] MEDIA

1: [14,5 9,5]
PRESE SORT LMEM MEDIA DOWN MULTI

Die Medianwerte betragen 14,5 für die erste Spalte und 9,5 für die zweite Spalte.

Vollständiges Erweitern und Zusammenfassen

Dieser Abschnitt enthält zwei Programme:

- *MULTI* wiederholt ein Programm solange, bis das Programm sein Argument nicht mehr verändert.
- *EXCO* ruft *MULTI* auf, um einen algebraischen Ausdruck vollständig zu erweitern und zusammenzufassen.

MULTI (Mehrfache Ausführung)

Gegeben sei ein Objekt und ein Programm, das auf dieses Objekt einwirkt. *MULTI* wendet das Programm wiederholt an, bis das Objekt nicht mehr verändert wird.

Argumente	Ergebnisse
2: <i>Objekt</i> 1: « <i>Programm</i> »	2: 1: <i>Resultierendes Objekt</i>

Verfahren

- **DO ... UNTIL ... END** (unbestimmte Schleife). Die **DO**-Anweisung enthält die Programmschritte, die wiederholt auszuführen sind; die **UNTIL**-Anweisung führt einen Test aus, um zu bestimmen, ob beide Anweisungen zu wiederholen sind (wenn das Ergebnis "falsch" ist) oder dieses Unterprogramm zu verlassen ist (wenn das Ergebnis "wahr" ist).
- **Programme als Argumente**. Obwohl Programme im allgemeinen einen Namen erhalten und anschließend durch Aufrufen ihres Namens ausgeführt werden, können sie auch im Stack abgelegt und als Argumente von anderen Programmen verwendet werden.

- **Verarbeiten von lokalen Variablen.** Das Programmargument, das wiederholt ausgeführt werden soll, ist in einer lokalen Variablen gespeichert. Es ist vorteilhaft, ein Objekt in einer lokalen Variablen zu speichern, wenn Sie vorher noch nicht wissen, wieviele Kopien Sie brauchen.

Auf Seite 103 können Sie nachlesen, daß ein Objekt, das in einer lokalen Variablen gespeichert ist, einfach nur in den Stack geholt wird, wenn die lokale Variable ausgewertet wird. Die tatsächliche Auswertung muß dann ausdrücklich durch EVAL erfolgen. *MULTI* verwendet den Namen der lokalen Variablen, um das Programmargument in den Stack zu holen und führt anschließend mit EVAL das Programm aus.

Programm:	Kommentar:
⌘	
→ P	Erzeugt eine lokale Variable <i>p</i> , die das Programm aus Ebene 1 enthält.
⌘	Leitet die definierende Prozedur (ein Programm) für die Struktur lokaler Variablen ein.
DO	Leitet die DO-Anweisung ein.
DUP	Dupliziert das Objekt, das sich nun in Ebene 1 befindet.
P EVAL	Wendet das Programm auf das Objekt an und liefert als Ergebnis eine neue Version des Objektes.
DUP	Dupliziert die neue Version des Objektes.
ROT	Verschiebt die alte Version nach Ebene 1
UNTIL	Leitet die UNTIL-Anweisung ein.
SAME	Testet, ob die alte Version und die neue Version identisch sind.

END	Schließt die UNTIL-Anweisung ab.
⌘	Schließt die definierende Prozedur ab.
⌘	Beendet das Programm.
[ENTER] [] MULTI [STO]	Speichert das Programm in <i>MULTI</i> .

Prüfsumme: # 34314d
Bytes: 56

Die Anwendung von *MULTI* wird im nächsten Programmbeispiel demonstriert.

EXCO (Vollständiges Erweitern und Zusammenfassen)

Gegeben sei ein algebraisches Objekt. *EXCO* führt *EXPAN* wiederholt aus, bis der algebraische Ausdruck sich nicht mehr ändert; anschließend wird *COLCT* ausgeführt und so lange wiederholt, bis sich wiederum der algebraische Ausdruck nicht mehr ändert. In einigen Fällen ist das Ergebnis eine Zahl.

Argument	Ergebnisse
1: 'Algebraischer Ausdruck '	1: 'Algebraischer Ausdruck '
1: 'Algebraischer Ausdruck '	1: z

Verfahren

- Unterprogramme. *EXCO* ruft das Programm *MULTI* zweimal auf. Es ist sinnvoller, das Programm *MULTI* nur einmal zu schreiben und einfach seinen Namen zweimal aufzurufen, als jeden Schritt in *MULTI* zweimal zu schreiben.

Erforderliche Programme

- *MULTI* (Seite 617) führt die Programme mehrfach aus, die *EXCO* als Argumente liefert.

Wählen Sie das Menü VAR und starten Sie das Programm.

VAR EXCO

1: '64*X^2+12*X*Y-77*X
*Z+25*Z^2'
GOIN MULT EXCO MIN1 MIN2 NAME

Bei Ausdrücken mit einer großen Zahl von Summenprodukten oder mit Potenzen muß EXPAN eventuell sehr oft wiederholt werden; das führt zu einer langen Ausführungszeit von EXCO.

Bestimmung des größten oder kleinsten Elementes in einem Feld

Dieser Abschnitt enthält zwei Programme, mit denen Sie das kleinste oder das größte Element eines Feldes ermitteln können:

- MNX verwendet eine (unbestimmte) DO ... UNTIL ... END Schleife.
- MNX2 verwendet eine (bestimmte) FOR ... NEXT Schleife.

MNX (Bestimmung des kleinsten oder des größten Elementes in einem Feld — Verfahren 1)

Gegeben ist ein Feld im Stack. MNX ermittelt das kleinste oder das größte Element in dem Feld.

Argumente	Ergebnisse
1: [[Feld]]	2: [[Feld]] 1: z (Größtes Element im Feld)
1: [[Feld]]	2: [[Feld]] 1: z (Kleinstes Element im Feld)

Verfahren

- DO ... UNTIL ... END (unbestimmte Schleife). Die DO-Anweisung enthält Befehle, die die Elemente sortieren. In der UNTIL-Anweisung wird ein Systemflag getestet und festgestellt, ob weitersortiert werden muß.

- Benutzer- und Systemflags zur logischen Steuerung:
 - *Benutzerflag 10* gibt die Sortierreihenfolge an. Ist Flag 10 gesetzt, bestimmt *MAX* das größte Element; wenn Flag 10 rückgesetzt ist, ermittelt es das kleinste Element. Sie legen den Zustand von Flag 10 am Anfang des Programms fest.
 - *System-Flag -64*, das Anzeigeflag für einen Indexüberlauf, bestimmt, wann das Sortieren beendet werden soll. Solange das Flag -64 rückgesetzt ist, wird die Sortierschleife weiter durchlaufen. Wenn der durch GETI verwendete Index durch den Überlauf wieder zum ersten Element des Feldes zurückkehrt, wird das Flag -64 *automatisch* gesetzt, und das Durchlaufen der Sortierschleife ist beendet.
- Eingebettete Verzweigung. Eine Verzweigungsstruktur IF ... THEN ... END ist in der Schleifenstruktur DO ... UNTIL ... END eingebettet — sie bestimmt, ob das bisher gültige größte oder kleinste Element erhalten bleiben soll oder ob das gerade untersuchte Element das neue größte oder kleinste Element wird.
- Benutzerspezifisches Menü als Unterstützung bei der Auswahl. *MAX* stellt ein benutzerspezifisches Menü dar; damit können Sie wählen, ob Sie das kleinste oder das größte Element bestimmen wollen. Taste 1 mit der Bezeichnung `MAX` setzt Flag 10. Taste 2 mit der Bezeichnung `MIN` setzt Flag 10 zurück.
- Logische Funktion. *MAX* führt XOR (*Exklusiv-ODER*) aus, um das Ergebnis des Vergleichs und das Flag 10 miteinander zu verknüpfen.

Programm:

```
⌘
```

```
{
```

```
{ "MAX"
⌘ 10 SF CONT ⌘ }
{ "MIN"
⌘ 10 CF CONT ⌘ }
```

Kommentar:

Beginnt die Liste, die zur Erzeugung des temporären Auswahlmenüs dient.

Die Menütaste `MAX` setzt Flag 10 und setzt das Programm fort.

Die Menütaste `MIN` setzt Flag 10 zurück und setzt das Programm fort.

<code>}</code>	Beendet die Liste für das temporäre Auswahlmenü.
<code>TMENU</code> "Sort for MAX or MIN?" <code>PROMPT</code>	Zeigt das temporäre Menü sowie eine Aufforderung an den Bediener an.
<code>1 GETI</code>	Nimmt das erste Element des Feldes.
<code>DO</code>	Beginnt die DO-Schleife.
<code>ROT ROT</code>	Überträgt den Index und das Feld in die Ebenen 1 und 2.
<code>GETI 4 ROLL DUP2</code>	Nimmt das neue Feldelement, verschiebt dann das zuletzt bestimmte kleinste oder größte Feldelement von Ebene 4 nach Ebene 1 und dupliziert anschließend beide Elemente.
<code>IF</code>	Leitet die Verzweigung ein.
<code>> 10 FS? XOR</code>	Testet gemeinsam das Vergleichsergebnis und den Zustand von Flag 10.
<code>THEN</code>	Wenn das neue Element entweder kleiner als das zuletzt bestimmte größte oder größer als das zuletzt bestimmte kleinste Element ist ...
<code>SWAP</code>	... wird das neue Element durch Vertauschen nach Ebene 1 gebracht.
<code>END</code>	Schließt die innere bedingte Struktur (Verzweigung) ab.
<code>DROP</code>	Bewahrt das zuletzt bestimmte kleinste oder größte Element auf und entfernt das andere Element vom Stack.

UNTIL

Leitet die UNTIL-Anweisung ein.

-64 FS?

Testet, ob Flag -64 gesetzt ist. Ist Flag -64 nicht gesetzt, wird die DO-Anweisung erneut ausgeführt.

END

Ist Flag -64 gesetzt, wird der Schleifendurchlauf beendet.

SWAP DROP 0 MENU

Bringt den Index nach Ebene 1 und entfernt ihn anschließend vom Stack. Ruft das letzte Menü wieder auf.

»

ENTER **□** MNX **STO**

Speichert das Programm in MNX.

Prüfsumme: # 57179d

Bytes: 210,5

Beispiel. Bestimmen Sie das größte Element der folgenden Matrix:

$$\begin{bmatrix} 12 & 56 \\ 45 & 1 \\ 9 & 14 \end{bmatrix}$$

Geben Sie die Matrix ein.

▶ **MATRIX**
12 **ENTER** 56 **ENTER** **▼**
45 **ENTER** 1 **ENTER**
9 **ENTER** 14 **ENTER**
ENTER

```
1: [[ 12 56 ]
    [ 45 1 ]
    [ 9 14 ] ]
MNX  CST  DL  SORT  BDISP  FIB1
```

Wählen Sie das Menü VAR und starten Sie MNX.

VAR MNX

```
Sort for MAX or MIN?
2:
1: [[ 12 56 ]
    [ 45 1 ]
    [ 9 14 ] ]
MNX  MIN
```

Bestimmen Sie das größte Element.

MAX

```
2: [[ 12 56 ] [ 45 1...
1:                                     56
MIN: 001 01 0001 0001 0001
```

MNX2 (Bestimmung des größten oder des kleinsten Elementes in einem Feld — Verfahren 2)

Gegeben ist ein Feld im Stack. *MNX2* bestimmt das größte oder das kleinste Element in dem Feld. *MNX2* verwendet einen anderen Weg als *MNX*; es führt OBJ→ aus, um das Feld in einzelne Elemente zu zerlegen, legt sie im Stack ab und untersucht sie dort, anstatt mit GETI schrittweise das Feld zu durchlaufen.

Argumente	Ergebnisse
1: [[Feld]]	2: [[Feld]] 1: z (Größtes Element im Feld)
1: [[Feld]]	2: [[Feld]] 1: z (Kleinstes Element im Feld)

Verfahren

- FOR ... NEXT (bestimmte Schleife). Der Anfangswert des Zählers ist 1. Der Endwert des Zählers lautet *nm* - 1, wobei *nm* die Anzahl der Elemente im Feld ist. Die Schleifenanweisung enthält die Befehle, die die Elemente sortieren.
- Benutzerflag zur logischen Steuerung. Das *Benutzer-Flag* 10 steuert die Sortierreihenfolge: Ist Flag 10 gesetzt, bestimmt *MNX2* das größte Element; wenn Flag 10 rückgesetzt ist, wird das kleinste Element ermittelt. Sie legen den Zustand von Flag 10 am Anfang des Programms fest.
- Eingebettete Verzweigung. Eine bedingte Verzweigung IF ... THEN ... END ist in die Schleife FOR ... NEXT eingebettet — sie bestimmt ob das bisher gültige kleinste oder größte Element erhalten bleiben soll oder ob das gerade untersuchte Element das neue kleinste oder größte Element wird.

- Logische Funktion. *MNX2* führt XOR (*Exklusiv-ODER*) aus, um das Ergebnis des Vergleichs und das Flag 10 miteinander zu verknüpfen.
- Benutzerspezifisches Auswahlmenü. *MNX2* stellt ein benutzer-spezifisches Menü dar; damit können Sie wählen, ob das größte oder das kleinste Element ermittelt werden soll. Taste 1 mit der Bezeichnung `MAX` setzt Flag 10. Taste 2 mit der Bezeichnung `MIN` setzt Flag 10 zurück.

Programm:

Kommentar:

«

{

```
{ "MAX"
« 10 SF CONT » }
{ "MIN"
« 10 CF CONT » }
```

}

```
TMENU
"Sort for MAX or MIN?"
PROMPT
```

DUP

OBJ→

1

SWAP OBJ→

Beginnt die Liste, die zur Erzeugung des temporären Auswahlmenüs dient.

Die Menütaste `MAX` setzt Flag 10 und setzt das Programm fort. Die Menütaste `MIN` setzt Flag 10 zurück und setzt das Programm fort.

Beendet die Liste für das Auswahlmenü.

Zeigt das temporäre Menü sowie eine Aufforderung an den Bediener an.

Dupliziert das Feld.

Legt die einzelnen Feldelemente in den Ebenen 2 bis einschließlich $nm + 1$ sowie die Liste mit n und m in Ebene 1 ab.

Legt den Anfangswert des Zählers fest.

Zerlegt auch die Liste in einzelne Elemente im Stack.

DROP * 1 -	Entfernt die Länge der Liste vom Stack und berechnet anschließend den Endwert des Zählers ($nm - 1$).
FOR n	Beginnt die Schleife FOR ... NEXT.
DUP2	Sichert die zu untersuchenden Feldelemente (beim ersten Durchlauf die beiden letzten). Macht das letzte Feldelement zum aktuell gültigen größten oder kleinsten Element.
IF	Leitet die Verzweigung ein.
> 10 FS? XOR	Testet gemeinsam das Vergleichsergebnis und den Zustand von Flag 10.
THEN	Wenn das neue Element entweder kleiner als das zuletzt bestimmte größte oder größer als das zuletzt bestimmte kleinste Element ist, ...
SWAP	... wird das neue Element durch Vertauschen nach Ebene 1 gebracht.
END	Schließt die innere bedingte Struktur ab.
DROP	Bewahrt das zuletzt bestimmte kleinste oder größte Element auf und entfernt das andere Element vom Stack.
NEXT	Beendet die Schleife FOR ... NEXT.
Ø MENU	Ruft das letzte Menü wieder auf.

»

[ENTER] **[]** MNX2 **[STO]**

Speichert das Programm in MNX2.

Prüfsumme: # 12277d

Bytes: 200,5

Beispiel. Verwenden Sie MNX2, um das kleinste Element in der Matrix aus dem vorigen Beispiel zu bestimmen:

$$\begin{bmatrix} 12 & 56 \\ 45 & 1 \\ 9 & 14 \end{bmatrix}$$

Geben Sie die Matrix ein.

[>] **[MATRIX]**
12 **[ENTER]** 56 **[ENTER]** **[v]**
45 **[ENTER]** 1 **[ENTER]**
9 **[ENTER]** 14 **[ENTER]**
[ENTER]

```
1: [[ 12 56 ]
    [ 45 1 ]
    [ 9 14 ]]
MIN  CST  01  SORT  BEEP  FIB
```

Wählen sie das Menü VAR und starten Sie MNX2.

[VAR] MNX2

```
Sort for MAX or MIN?
2:
1: [[ 12 56 ]
    [ 45 1 ]
    [ 9 14 ]]
MAX  MIN
```

Bestimmen Sie das kleinste Element.

[MIN]

```
2: [[ 12 56 ] [ 45 1...
1:
MEMO DOWN MULTI ERGZ MNRE NAME
```

Prüfung von Programmargumenten

Mit den beiden Dienstprogrammen in diesem Abschnitt können Sie prüfen, ob das Argument für ein Programm den richtigen Objekttyp besitzt.

- *NAMES* prüft, ob ein Listenargument genau zwei Namen enthält.

- *VFY* prüft, ob das Argument entweder ein Name oder eine Liste ist, die genau zwei Namen enthält. Es ruft *NAMES* auf, wenn das Argument eine Liste ist.

Sie können diese Dienstprogramme ändern, wenn Sie andere Objekttypen und Objektinhalte prüfen wollen.

NAMES (Enthält die Liste genau zwei Namen?)

Ist das Argument für ein Programm eine Liste (wie mit *VFY* festgestellt), überprüft *NAMES*, ob die Liste genau zwei Namen enthält. Enthält die Liste nicht genau zwei Namen, wird eine Fehlermeldung im Statusbereich angezeigt und das Program wird abgebrochen.

Argumente	Ergebnisse
1: { Gültige Liste }	1:
1: { Ungültige Liste }	Fehlermeldung im Statusbereich 1:

Verfahren

- Verschachtelte bedingte Strukturen. Die äußere bedingte Verzweigung testet, ob die Liste zwei Objekte enthält. Sind in der Liste zwei Objekte vorhanden, testet die innere Verzweigung, ob diese beiden Objekte Namen sind.
- Logische Funktionen. *NAMES* verwendet den Befehl AND in der inneren bedingten Struktur, um zu ermitteln, ob *beide* Objekte Namen sind und den Befehl NOT, um eine Fehlermeldung anzuzeigen, wenn nicht beide Objekte Namen sind.

Programm:

Kommentar:

⌘

IF

Leitet die äußere
Verzweigungsstruktur
IF...THEN...ELSE...END ein.

OBJ→	Legt die n Objekte der Liste in den Ebenen 2 bis einschließlich $(n + 1)$ und die Länge der Liste n in Ebene 1 ab.
DUP	Dupliziert Länge der Liste.
2 SAME	Testet, ob die Länge der Liste 2 ist.
THEN	<i>Ist die Länge der Liste 2 ...</i>
DROP	..., so werden die Objekte auf die Ebenen 1 und 2 verschoben.
IF	Leitet die innere Struktur IF...THEN...END ein.
TYPE 6 SAME	Testet, ob das erste Objekt ein Name ist. Wenn ja, wird das Ergebnis "wahr" (1) ausgegeben, wenn nicht, das Ergebnis "falsch" (0).
SWAP TYPE 6 SAME	Verschiebt das zweite Objekt nach Ebene 1 und testet anschließend, ob dieses Objekt ein Name ist.
AND	Sind beide Ergebnisse "wahr", wird wiederum das Ergebnis "wahr" (1) ausgegeben. Wenn ein Ergebnis oder beide Ergebnisse "falsch" sind, wird das Ergebnis "falsch" (0) ausgegeben.
NOT	Invertiert das Ergebnis.
THEN	Ist das invertierte Ergebnis "wahr" (wenn also nicht beide Objekte Namen sind)...
"List needs two names" DOERR	... erscheint eine Fehlermeldung, und das Programm wird abgebrochen.

END	Schließt die innere bedingte Struktur ab.
ELSE	Ist die Länge der Liste nicht 2 ...
DROPN "Illegal list size" DOERR	... wird die Länge der Liste vom Stack entfernt, es erscheint eine Fehlermeldung, und das Programm wird abgebrochen.
END	Schließt die äußere bedingte Struktur ab.

✳

[ENTER] **[]** NAMES **[STO]** Speichert das Programm in *NAMES*.

Prüfsumme: # 40666d

Bytes: 141,5

Die Anwendung von *NAMES* wird im Programm *VFY* demonstriert.

VFY (Überprüfung von Argumenten für Programme)

Gegeben ist ein Argument im Stack. *VFY* testet, ob das Argument entweder ein Name oder eine Liste ist, die genau zwei Namen enthält.

Argumente	Ergebnisse
1: 'Name'	1: 'Name'
1: { Gültige Liste }	1: { Gültige Liste }
1: { Ungültiges Objekt }	Fehlermeldung im Statusbereich 1: { Ungültige Liste }
1: Ungültiges Objekt	Fehlermeldung im Statusbereich 1: Ungültiges Objekt

Verfahren

- Dienstprogramme. *VFY* ist für sich allein von geringem Nutzen. Es kann allerdings (mit kleinen Änderungen) von anderen Programmen aufgerufen werden, um zu testen, ob bestimmte Objekttypen gültige Argumente sind.
- CASE ... END (Case-Struktur). *VFY* verwendet eine Case-Struktur, um zu bestimmen, ob das Argument eine Liste oder ein Name ist.
- Strukturierte Programmierung. Ist das Argument eine Liste, ruft *VFY* das Unterprogramm *NAMES* auf, um festzustellen, ob es sich um eine gültige Liste handelt.
- Struktur lokaler Variablen. *VFY* speichert sein Argument in einer lokalen Variablen, so daß es, falls erforderlich, von *NAMES* übernommen werden kann.
- Logischer Operator. *VFY* verwendet NOT, um eine Fehlermeldung anzuzeigen.

Erforderliche Programme.

- *NAMES* (Seite 628) überprüft, ob eine Liste genau zwei Namen enthält.

Programm:

Kommentar:

⌘

DUP

Sichert das Originalargument.

D-TAG

Für den folgenden Test werden alle vorhandenen Markierungen entfernt.

→ arg

Speichert das Argument in der lokalen Variablen *arg*.

⌘

Leitet die definierende Prozedur (ein Programm) für die Struktur lokaler Variablen ein.

CASE

Leitet die Case-Struktur ein.

arg TYPE 5 SAME

Testet, ob das Argument eine Liste ist.

THEN	Wenn das Argument eine Liste ist ...
arg NAMES	... wird es in den Stack zurückgegeben, und es wird <i>NAMES</i> aufgerufen, um zu prüfen, ob die Liste gültig ist.
END	Schließt den ersten Fall ab (War das Ergebnis "wahr", wird die Case-Struktur verlassen. War es "falsch", so wird zum nächsten Fall gesprungen)
arg TYPE 6 SAME NOT	Testet, ob das Argument ein Name ist und invertiert anschließend das Testergebnis.
THEN	Ist das Argument <i>kein</i> Name (und keine Liste) ...
"Not name or list" DOERR	..., so wird eine Fehlermeldung angezeigt und das Programm abgebrochen.
END	Beendet die zweite Fallunterscheidung.
END	Beendet die Case-Struktur.
»	Beendet die definierende Prozedur.
»	
[ENTER] [] VFY [STO]	Speichert das Programm in VFY.

Prüfsumme: # 14621d
Bytes: 135,5

Beispiel. Teil 1. Führen Sie VFY aus, um die Gültigkeit des Namens PAT zu testen.

Bringen Sie den Namen PAT in den Stack. Wählen sie das Menü VAR und starten Sie VFY.

[] PAT **[ENTER]**
[VAR] VFY

1: 'PAT'
 VFY BER SINTP SETTS TSH PIE

Das Argument ist gültig und wird einfach in den Stack zurück übertragen.

Teil 2. Benutzen Sie *VFY*, um die Gültigkeit des Listenargumentes { *PAT DIANA TED* } zu prüfen..

Geben Sie die Namen *DIANA* und *TED* in den Stack ein. Wandeln Sie die Namen im Stack in eine Liste um.

DIANA [ENTER]

TED [ENTER]

3 [PRG] [OBJ] [LIST]

```
1: { PAT DIANA TED }  
[OBJ] [EQ] [ARR] [LIST] [STR] [TAG]
```

Starten Sie *VFY*. Da die Liste zu viele Namen enthält, erscheint die Fehlermeldung, und das Programm wird abgebrochen.

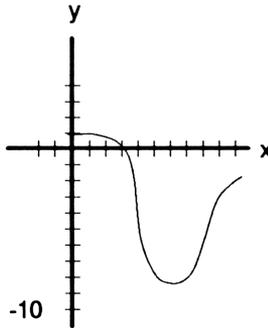
[VAR] VFY

```
Illegal list size  
4:  
3:  
2:  
1: { PAT DIANA TED }  
[MULTI] [EQD] [MIN1] [MIN2] [NAME] [VFY]
```

Bessel-Funktionen

Realteil und Imaginärteil der Bessel-Funktion $J_n(xe^{\frac{3\pi i}{4}})$ werden mit $Ber_n(x)$ und $Bei_n(x)$ bezeichnet. Bei $n = 0$ ergibt sich

$$Ber(x) = 1 - \frac{(x/2)^4}{2!^2} + \frac{(x/2)^8}{4!^2} - \dots$$



Die Funktion *BER* berechnet $Ber(x)$ auf 12 Stellen genau.

Argumente	Ergebnisse
1: z	1: $Ber(z)$

Verfahren

- **Struktur lokaler Variablen.** *BER* besteht nur aus einer Struktur lokaler Variablen und akzeptiert daher wie eine benutzerdefinierte Funktion numerische oder symbolische Argumente vom Stack oder in algebraischer Syntax. Da *BER* mit einer FOR ... STEP-Schleife arbeitet, ist seine definierende Prozedur ein *Programm*. (Schleifen sind in algebraischen Ausdrücken nicht zulässig.) Aus diesem Grund ist *BER*, anders als eine benutzerdefinierte Funktion, nicht differenzierbar.

- **FOR ... STEP-Schleife** (Bestimmte Schleife mit Zähler).
Aufeinanderfolgende Glieder in der Reihe werden mit Hilfe einer zählergesteuerten Schleife berechnet. Wenn das letzte Glied innerhalb der 12-stelligen Genauigkeit des Taschenrechners nicht mehr zu einer Änderung des Wertes der Reihe führt, wird das Durchlaufen der Schleife abgebrochen. Der Zählerendwert ($9,0 \times 10^{499}$) garantiert, daß eine genügend große Zahl von Gliedern berechnet wird.
- **Eingebettete bedingte Struktur**. Die Verzweigung **IF ... THEN ... ELSE ... END** innerhalb der bestimmten Schleife stellt die Schrittweite n für den Schleifenzähler ein. Solange sich der zuletzt berechnete Wert der Reihe von dem vorhergehenden Reihenwert unterscheidet, wird die Schrittweite n auf 2 gesetzt. Wenn der letzte Wert der Reihe *gleich* dem vorhergehenden Wert *ist*, wird als Schrittweite ein Wert gewählt, der größer als der Zählerendwert ist. Damit endet die bestimmte Schleife. Im wesentlichen bewirkt die eingebettete Verzweigung, daß die äußere Schleife wie eine (unbestimmte) Schleife **DO ... UNTIL ... END** wirkt.

Programm:

Kommentar:

<pre> « → × « 1 2 9.E499 FOR j DUP '(-1)^(j/2)* (x/2)^(2*j) /SQ(j!)' EVAL </pre>	<pre> Erzeugt die lokale Variable x. Leitet die definierende Prozedur (ein Programm) für die Struktur lokaler Variablen ein. Schreibt den ersten Ausdruck der Reihe. Stellt den Zähler für die Schleife FOR ... STEP ein. Beginnt die Schleife. Sichert den zuletzt berechneten Wert der Reihe (beim ersten Durchlauf 1). Berechnet das nächste Glied der Reihe. </pre>
--	---

+	Addiert das nächste Glied und den zuletzt berechneten Wert der Reihe, um einen neuen Reihenwert zu erhalten.
IF	Leitet die bedingte Struktur ein.
DUP ROT ≠	Testet, ob der neue Wert der Reihe sich von dem vorhergehenden Wert unterscheidet.
THEN	Unterscheiden sich der neue und der alte Wert voneinander ...
2	..., so wird $n = 2$ gesetzt.
ELSE	Sind der neue und der vorhergehende Ausdruck gleich (mit 12-stelliger Genauigkeit) ...
9.1E499	..., so wird $n = 9,1E499$ gesetzt.
END	Beendet die bedingte Struktur.
STEP	Legt die Schrittweite in Abhängigkeit vom Ergebnis der bedingten Struktur fest.
* *	Beendet die definierende Prozedur.
[ENTER] [] BER [STO]	Speichert das Programm in <i>BER</i> .

Prüfsumme: # 872d
Bytes: 148

Beispiel. Berechnen Sie Ber(3).

[VAR]
3 BER

1: -,2213802496
BER

Berechnen Sie Ber(2) in algebraischer Syntax.

BER ← () 2
EVAL

1: ,751734182714
BER

Bewegte Darstellung einer Folge von Taylor-Polynomen

Dieser Abschnitt enthält drei Programme, die mit Grafikobjekten arbeiten, um eine Folge von Taylor-Polynomen für die Sinus-Funktion darzustellen.

- *SINTP* erzeugt eine Sinuskurve und speichert ihre grafische Darstellung in einer Variablen.
- *SETTS* überlagert der grafischen Darstellung einer Sinuskurve aus *SINTP* die Darstellungen einer Folge von Taylor-Polynomen und speichert jedes Grafikobjekt in einer Liste.
- *TSA* stellt der Reihe nach jedes einzelne Grafikobjekt aus der von *SETTS* gebildeten Liste dar.

Erzeugen einer Sinuskurve und Umwandlung in ein Grafikobjekt

SINTP erzeugt eine Sinuskurve, überträgt die Darstellung als Grafikobjekt in den Stack und speichert dieses Grafikobjekt in einer Variablen.

Argument	Ergebnisse
1:	1:

Verfahren

- Verwendung der PLOT-Befehle in Programmen, um Grafikobjekte zu erzeugen und darzustellen.

Programm:**Kommentar:**

«

```
'X' PURGE
'SIN(X)' STEQ
```

Macht X zu einer formalen Variablen und speichert den Ausdruck für $\sin x$ in EQ .

```
-2 2 YRNG
```

Wählt den Anzeigebereich der y -Achse.

```
ERASE DRAW
```

Löscht $PICT$ und stellt anschließend den Ausdruck grafisch dar.

```
PICT RCL 'SINT' STO
```

Legt das resultierende Grafikobjekt im Stack ab und speichert es in $SINT$.

»

```
[ENTER] [ ] SINTP [STO]
```

Speichert das Programm in $SINTP$.

Pürfsumme: # 61373d

Bytes: 78,5

Überlagerung einer Folge von Taylor-Polynomen

$SETTS$ überlagert einer Sinuskurve eine Folge von Taylor-Polynomen und speichert jedes Grafikobjekt in einer Liste.

Argumente	Ergebnisse
1:	1:

Verfahren

- Strukturierte Programmierung. $SETTS$ ruft $SINTP$ zur Darstellung einer Sinuskurve und ihrer Umwandlung in ein Grafikobjekt.
- (Bestimmte) Schleife FOR ... STEP. $SETTS$ berechnet aufeinander folgende Taylor-Polynome für die Sinusfunktion in einer bestimmten Schleife. Der Wert des Schleifenzählers bestimmt die Ordnung jedes einzelnen Polynoms.

- Verwendung von PLOT-Befehlen in Programmen. *SETTS* stellt jedes einzelne Taylor-Polynom grafisch dar.
- Bearbeitung von Grafikobjekten. *SETTS* wandelt jede einzelne grafische Darstellung der Taylor-Polynome in ein Grafikobjekt um. Dann führt es die Funktion GOR (*Grafisches ODER*) aus, um jedes einzelne Grafikobjekt mit der in *SINT* gespeicherten Sinuskurve zu überlagern. Dabei entstehen neun neue Grafikobjekte, von denen jedes einzelne die Überlagerung eines Taylor-Polynoms mit der Sinuskurve darstellt. *SETTS* faßt die neun neuen Grafikobjekte und das Grafikobjekt mit der Sinuskurve in einer Liste zusammen.

Programm:

Kommentar:

⌘

SINTP

Plottet eine Sinuskurve und speichert das Grafikobjekt in *SINT*.

17 1 FOR x

Für jeden einzelnen Wert der lokalen Variablen x ...

x 'X' DUP

SIN SWAP ROT TAYLR

STEQ ERASE DRAW

... wird das Taylor-Polynom der Sinuskurve geplottet (wobei x die Ordnungszahl des Polynoms ist).

PICT RCL SINT +

Überträgt die Darstellung als Grafikobjekt zurück auf den Stack und führt die Operation + aus, um die Taylorreihe der in *SINT* gespeicherten Sinuskurve zu überlagern.

-2 STEP

Vermindert den Wert des Schleifenzählers (die Ordnungszahl des Taylor-Polynoms) um 2 und durchläuft die Schleife erneut.

```
SINT 10 →LIST
'TSL' STO
```

Legt das Grafikobjekt der Sinuskurve im Stack ab und stellt anschließend dieses Grafikobjekt und die neun Grafikobjekte, die in der Schleife FOR ... STEP entstanden sind, in einer Liste zusammen. Speichert die Liste in *TSL*.

✳

```
[ENTER] [ ] SETTS [STO]
```

Speichert das Programm in *SETTS*.

Prüfsumme: # 5841d
Bytes: 136,5

Bewegte Darstellung von Taylor-Polynomen

Mit *TSA* wird der Reihe nach jedes einzelne Grafikobjekt angezeigt, das mit *SETTS* erzeugt worden ist.

Argumente	Ergebnisse
1:	1:

Verfahren

- Übergabe einer globalen Variablen. Weil *SETTS* längere Zeit zur Ausführung benötigt (ungefähr sechs Minuten), wird *SETTS* nicht von *TSA* aufgerufen. Stattdessen müssen Sie zunächst *SETTS* laufen lassen, um die globale Variable *TSL* anzulegen, die die Liste der grafischen Objekte enthält. *TSA* verwendet einfach diese globale Variable, um die Liste im Stack abzulegen.
- (Bestimmte Schleife) FOR ... NEXT. *TSA* führt eine bestimmte Schleife aus, um nacheinander jedes einzelne grafische Objekt aus der Liste anzuzeigen.

Programm:**Kommentar:**

⌘

TSL

Legt die Liste *TSL* im Stack ab.

OBJ→

Zerlegt die Liste in die 10 Grafikobjekte und legt diese zusammen mit der Anzahl der Objekte im Stack ab.

1 SWAP FOR s

Für *s* von 1 bis 10 ...

ERASE →LCD

1 WAIT

... wird das Display gelöscht, das Grafikobjekt in Ebene 1 in ein Bild umgewandelt und eine Sekunde lang angezeigt.

NEXT

⌘

[ENTER] [] TSA [STO]

Speichert das Programm in *TSA*.**Prüfsumme:** # 39562d**Bytes:** 51

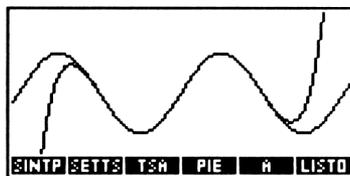
Beispiel. Verwenden Sie *SETTS* und *TSA*, um eine Folge von Taylor-Polynomen zur näherungsweisen Darstellung der Sinusfunktion zu erzeugen und nacheinander darzustellen.

Wählen Sie den Winkelmodus Radiant. Verwenden Sie *SETTS*, um die Liste der Grafikobjekte anzulegen. *SETTS* benötigt ungefähr sechs Minuten für die Ausführung. Stellen Sie die einzelnen Grafiken nacheinander mit *TSA* dar. Die Abbildung unten zeigt *TSA* in Aktion.

[↩] [RAD] (falls erforderlich)

[VAR] SETTS

TSA



Anwendung von Statistikdaten und grafischen Darstellungen in Programmen

Das Programm *PIE* fordert Sie auf, Daten einer einzelnen Variablen einzugeben, speichert diese Daten in der Statistikmatrix *SDAT* und erzeugt ein beschriftetes Tortendiagramm, das den prozentualen Anteil jedes einzelnen Datenwertes an der Summe darstellt.

Argumente	Ergebnisse
1 :	1 :

Verfahren

- Verwendung von PLOT-Befehlen in Programmen. *PIE* benutzt *XRNG* und *YRNG*, um die Anzeigebereiche der *x*- und der *y*-Achse in Benutzerkoordinaten festzulegen, erzeugt mit *ARC* den Kreis und mit *LINE* die einzelnen Kreisabschnitte.
- Verwendung von Matrizen- und Statistik-Befehlen in Programmen.
- Bearbeitung von Grafikobjekten. *PIE* ruft *PICT* auf den Stack und führt *GOR* aus, um der grafischen Darstellung die Beschriftung jedes Anteils zu überlagern.
- (Bestimmte Schleife) *FOR ... NEXT*. Jeder Anteil am Kreisdiagramm wird in einer bestimmten Schleife berechnet, dargestellt und beschriftet.
- *CASE ... END*-Struktur. Um das Überschreiben des Kreises zu vermeiden, wird jede Beschriftung vom Halbierungspunkt des Kreisbogens eines Anteils verschoben. Die Größe der Verschiebung ist abhängig von der Position des Anteils im Kreis. Die Struktur *CASE ... END* ordnet einer Beschriftung eine bestimmte Verschiebung zu, die von der Position des Anteils abhängig ist.
- Sicherung des aktuellen Flag-Status des Taschenrechners. Bevor der Winkelmodus *Radiant* aktiviert wird, speichert *PIE* den aktuellen Flag-Status in einer lokalen Variablen und stellt diesen Status am Ende des Programms wieder her.
- Temporäres Menü für die Dateneingabe.

Programm:

⌘

RCLF → flags

⌘

RAD

(

("SLICE" Σ+)

()

("CLEAR" CLΣ)

() ()

("DRAW" CONT)

)

TMENU

"Key values into
SLICE, ■ DRAW
restarts program."
PROMPT

ERASE 1 131 XRNG
1 64 YRNG CLLCD

"Please wait.....■
Drawing Pie Chart"
1 DISP

Kommentar:

Holt den aktuellen Flag-Status in den Stack und speichert ihn in der Variablen *flags*.

Wählt den Winkelmodus Radiant

Beginnt die Liste, die zur Erzeugung des Eingabemenüs dient.

Definiert Taste 1. Taste 1 führt Σ+ aus, um jeden Datenpunkt in ΣDAT zu speichern.

Definiert die Tasten 2 und 3. Taste 3 löscht ΣDAT.

Definiert die Tasten 4, 5 und 6. Taste 6 mit der Bezeichnung DRAW im Menüfeld setzt die Programmausführung nach der Dateneingabe fort.

Schließt die Liste für das Eingabemenü.

Zeigt das temporäre Menü an.

Fordert zur Eingabe von Daten auf. Das ■ ist die Darstellung des Taschenrechners für das Zeichen ← (→←), nachdem das Programm im Stack abgelegt worden ist.

Löscht das aktuelle PICT und legt die Plotparameter fest.

Meldet, daß die Grafik erstellt wird.

(66,32) 20 0 6.28 ARC	Erzeugt mit ARC den Kreis.
PICT RCL →LCD	Stellt den leeren Kreis dar.
RCLΣ TOT /	Holt die Matrix mit den Statistikdaten in den Stack, berechnet die Summen und ermittelt die einzelnen absoluten Anteile.
DUP 100 *	Wandelt die absoluten Anteile in prozentuale Anteile um.
→ prcnts	Speichert die Prozentsatz-Matrix in <i>prcnts</i> .
⊗	
2 π →NUM * * 0	Multipliziert the Anteilmatrix mit 2π .
→ prop angle	Speichert die Anteile in <i>prop</i> und setzt den Anfangswert des <i>Winkels</i> auf 0.
⊗	
prop SIZE OBJ→ DROP SWAP	Legt Anfangswert und Endwert für die Schleife FOR...NEXT fest.
FOR ×	Beginnt die FOR-Anweisung.
(66,32) prop × GET 'angle' STO+	Legt den Kreismittelpunkt im Stack ab und holt den <i>xten</i> Wert aus der Anteilmatrix.
angle COS LASTARG SIN R→C 20 * OVER + LINE	Berechnet den Endpunkt und erzeugt die Linie für den <i>xten</i> Anteil.
PICT RCL	Holt <i>PICT</i> in den Stack.
angle prop × GET 2 / - DUP COS LASTARG SIN R→C 26 * (66,32) +	Um einen Anteil zu beschriften, wird der Halbierungspunkt des Kreisbogens eines Anteils berechnet.

```
SWAP DUP
CASE
```

Beginnt die CASE...END-Struktur,
die den Wert der Verschiebung für
die Beschriftung bestimmt.

```
1.5 ≤
```

```
THEN
```

Von 0 bis 1,5 rad ...

```
DROP
```

... wird die Beschriftung nicht
versetzt.

```
END
```

```
DUP 4.4 ≤
```

```
THEN
```

Von 1,5 bis 4,4 rad ...

```
DROP 15 -
```

... wird die Beschriftung um 15
Benutzereinheiten nach links
verschoben.

```
END
```

```
5 <
```

```
THEN
```

Von 4,4 bis 5 rad ...

```
(3,2) +
```

... wird die Beschriftung um 3
Einheiten nach rechts und um 2
Einheiten nach oben verschoben.

```
END
```

```
END
```

```
prcnts x GET
```

Holt den xten Wert aus der
Prozentsatz-Matrix.

```
1 RND
```

Rundet den Prozentsatz auf eine
Dezimalstelle.

```
→STR "% " +
```

Wandelt den Prozentsatz in eine
Zeichenkette um und fügt der
Zeichenkette ein % hinzu.

1 →GROB

Wandelt die Zeichenkette in ein Grafikobjekt um.

GOR DUP PICT STO

Fügt der grafischen Darstellung die Beschriftung hinzu und speichert die neue Darstellung.

→LCD

Zeigt das Diagramm mit dem neuen Anteil und der zugehörigen Beschriftung an.

NEXT

() PVIEW

Zeigt die fertige grafische Darstellung an.

»

»

flags STOF

Stellt den ursprünglichen Flag-Status wieder her.

» 2 MENU

Stellt das Menü VAR dar. (Beachten Sie, daß Sie zuerst [ATTN] drücken müssen, um die grafische Darstellung zu löschen.)

»

[ENTER] [] PIE [STO]

Speichert das Programm in *PIE*.

Prüfsumme: # 8706d

Bytes: 758,5

Beispiel. Das Obstangebot in einem Laden besteht aus 983 Orangen, 416 Äpfeln und 85 Bananen. Erstellen Sie ein Tortendiagramm, das den prozentualen Anteil der einzelnen Obstsorten am gesamten Obstbestand darstellt.

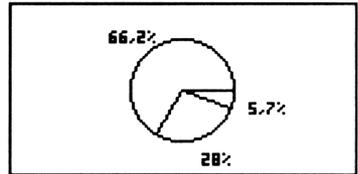
Starten Sie *PIE*.

VAR PIE

```
Key values into SLICE
DRAW restarts program.
4:
3:
2:
1:
SLICE CLEAR DRAW
```

Löschen Sie die zuletzt verwendeten statistischen Daten. (Der Bedienerhinweis verschwindet aus dem Anzeigefeld.) Geben Sie die neuen Daten ein und erstellen Sie das Tortendiagramm.

```
CLEAR
983 SLICE
416 SLICE
85 SLICE
DRAW
```



Bewegte Darstellung einer Grafik

Das Programm *WALK* zeigt eine Figur, die durch das Anzeigefeld spaziert. Die Bewegung dieser benutzerspezifischen Grafik wird dadurch erreicht, daß die Koordinate der Position der Figur in einer Schleifenstruktur ständig erhöht wird.

Argumente	Ergebnisse
1:	1:

Verfahren

- Verwendung einer benutzerspezifischen Grafik in einem Programm. (Beachten Sie, daß der Programmierer, bevor er das Programm schreibt, den vollen Informationsgehalt der grafischen Darstellung erhält, indem er das Bild *interaktiv* in der Grafikumgebung aufbaut und anschließend damit in die Befehlszeile zurückkehrt.)
- Bestimmte Schleife FOR...STEP, um die grafische Darstellung mit Leben zu erfüllen. Der Endwert für die Schleife lautet MAXR. Da der Zählerwert nicht größer als MAXR werden kann, wird die Schleife unbegrenzt oft durchlaufen.

Programm:

⌘

```
GROB 9 15 E300
140015001C001400E300
8000C110AA0094009000
4100220014102800
```

```
→ man
```

⌘

```
ERASE ( # 0d # 0d )
PVIEW
```

```
( # 0d # 25d )
PICT OVER man GXOR
```

```
5 MAXR FOR i
```

```
i 131 MOD R→B
```

```
# 25d 2 →LIST
```

```
PICT OVER man GXOR
```

```
PICT ROT man GXOR
```

```
5 STEP
```

⌘

⌘

Kommentar:

Grafische Darstellung der Figur in der Befehlszeile. (Beachten Sie, daß der hexadezimale Teil des grafischen Objektes eine fortlaufende ganze Zahl ist: E300. . . 2800. Die Zeilenumbrüche stellen *keine* Leerstellen dar.)

Erzeugt die lokale Variable *man*, die das grafische Objekt enthält.

Löscht *PICT* und zeigt es anschließend an.

Legt die erste Position im Stack ab und zeigt das erste Bild. Dadurch werden der Stack und *PICT* für die Schleife vorbereitet.

Beginnt die Schleife FOR...STEP, um die horizontalen Koordinaten unendlich oft zu erzeugen.

Berechnet die horizontale Koordinate für das nächste Bild.

Bestimmt eine feste vertikale Koordinate. Faßt die beiden Koordinaten in einer Liste zusammen.

Stellt das neue Bild dar; die Koordinaten stehen noch im Stack.

Löscht das alte Bild aus dem Anzeigefeld und entfernt seine Koordinaten vom Stack.

Erhöht die horizontale Koordinate um 5.

[ENTER] **[]** WALK **[STO]**

Speichert das Programm in *WALK*.

Prüfsumme: # 4342d

Bytes: 236,5

Beispiel. Schicken Sie die Figur auf einen langen Spaziergang.

Wählen Sie das Menü VAR und starten Sie *WALK*.

[VAR] **[WALK]**

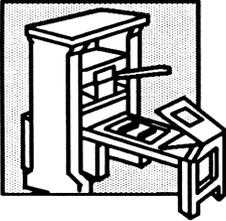


Wenn die Figur müde wird, drücken Sie **[ATTN]**, um sie nach Hause zu bringen (und um das Programm zu beenden).

Teil 5

Drucken, Datenübertragung und Einsteckmodule

Drucken



In diesem Kapitel ist beschrieben, wie Sie Ihren HP 48 mit einem Infrarotdrucker HP 82240B, mit einem Infrarotdrucker HP 82240A oder mit solchen Druckern verwenden können, die eine serielle Schnittstelle besitzen.

Drucken mit einem Drucker HP 82240B

Über die Infrarot-Schnittstelle können Sie Informationen von Ihrem HP 48 an den Infrarotdrucker HP 82240B übertragen. In Ihrem Druckerhandbuch ist ausführlich beschrieben, wie der Drucker zu bedienen ist und wie der Drucker vom HP 48 angesteuert werden kann.

Druckbefehle PRINT

Tasten	Programmierbarer Befehl	Beschreibung
   	PR1	Wenn Sie  und  gleichzeitig drücken und anschließend wieder loslassen, wird die aktuelle Anzeige gedruckt. Druckt das Objekt in Ebene 1.
  :		
	PR1	Druckt das Objekt in Ebene 1.
	PRST	Druckt alle Objekte aus dem Stack und beginnt dabei mit dem Objekt in der höchsten Ebene.
PRSTC	PRSTC	Druckt alle Objekte aus dem Stack in kompakter Form und beginnt dabei mit dem Objekt in der höchsten Ebene.
	PRLCD	Druckt die aktuelle Anzeige.
	PRVAR	Sucht den aktuellen Programmzweig für die angegebenen Variablen, und druckt Namen und Inhalt jeder Variablen. Die Variablen werden entweder durch einen Namen gekennzeichnet oder in einer Liste aufgeführt. Name oder Liste stehen in Ebene 1.
	CR	Veranlaßt den Drucker dazu, einen Druckkopfrücklauf/Zeilenvorschub auszuführen und dabei den Inhalt des Druckerpuffers zu drucken, falls vorhanden.
DELAY	DELAY	Wählt die Verzögerungszeit $\leq 6,9$ Sekunden zwischen der Übertragung einzelner Datenzeilen an den Drucker.

Druckbefehle PRINT (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
OLDPR	OLDPRT	Paßt den Zeichensatz des HP 48 an den Infrarotdrucker HP 82240A an.

Druckformate

Mehrzeilige Objekte können im *mehrzeiligen* Format oder im *kompakten* Format gedruckt werden. Das mehrzeilige Druckerformat ist ähnlich dem mehrzeiligen Anzeigeformat; allerdings mit den folgenden Ausnahmen:

- Zeichenfolgen und Namen, die mehr als 24 Zeichen lang sind, werden in der nächsten Druckerzeile fortgesetzt.
- Realteil und Imaginärteil von komplexen Zahlen werden in getrennten Zeilen gedruckt, wenn sie nicht in dieselbe Zeile passen.
- Datentabellen werden mit einer numerierten Überschrift für jede Zeile und mit einer Spaltennummer vor jedem Element gedruckt. Zum Beispiel würde die 2×3 -Matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

in dieser Form gedruckt:

Feldgröße
↓

Array (2 3)

Zeilennummer → Row 1

Spaltennummer { 1] 1
2] 2
3] 3

Row 2

1] 4
2] 5
3] 6

Das kompakte Druckerformat ist dasselbe wie das kompakte Anzeigeformat: Mehrzeilige Dateninhalte werden verkürzt und erscheinen nur in einer Zeile.

Mit dem Befehl PRSTC wird der Stack in kompakter Form gedruckt. Mit allen anderen Druckbefehlen werden die Daten mehrzeilig ausgedruckt.

Grunddruckbefehle

Druck des Anzeigeinhalts. Wenn Sie einen beliebigen Anzeigeinhalt drucken wollen, ohne das Menü PRINT zu verwenden, sind folgende Bedienungsschritte auszuführen: *

1. Drücken Sie **[ON]** und halten Sie die Taste gedrückt.
2. Drücken Sie **[MTH]** (Bezeichnung "PRINT" oberhalb der Taste).
3. Lassen Sie **[ON]** los.

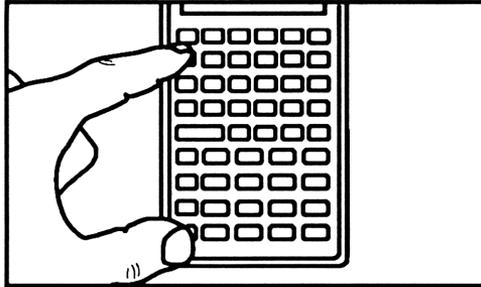


Hinweis

Verbrauchte Batterien mit zu geringer Spannung führen dazu, daß nach dem Starten des Druckvorgangs mit den Tasten **[ON]** **[MTH]** beim Drucken ständig Fehler auftreten. Wenn Sie solche Fehler feststellen, müssen Sie die

Batterien Ihres Rechners austauschen.

* Diese Tastendrucke verwenden die zuletzt gewählte DELAY-Einstellung. Wenn Sie Druckdaten zum seriellen Anschluß ausgeben, um grafische Daten auf Ihrem Drucker zu erfassen, müssen Sie, bevor Sie diese Tasten betätigen, den seriellen Anschluß mit dem Befehl OPENIO geöffnet werden haben.



Mit dem Befehl PRLCD (  PRLCD) können Sie ebenfalls den aktuellen Anzeigehalt drucken.

Druck des Inhalts von Ebene 1 des Stacks. PR1 (  PR1) druckt den Inhalt von Ebene 1 im Mehrzeilenformat des Druckers. Alle Objekte mit Ausnahme von Zeichenfolgen werden mit ihren kennzeichnenden Begrenzungszeichen gedruckt. Zeichenfolgen werden ohne ihre " Begrenzer gedruckt. PR1 kann auch durch Drücken von   ausgeführt werden.

Druck des Stacks. PRST (  PRST) druckt alle Objekte des Stacks im Mehrzeilenformat des Druckers, beginnend mit dem Objekt der obersten Ebene (mit Ausnahme von grafischen Objekten, die genauso gedruckt werden, wie sie in der Anzeige erscheinen).

PRSTC (  PRSTC) druckt alle Dateninhalte des Stacks im kompakten Format des Druckers, beginnend mit dem Dateninhalt der obersten Ebene.

Druck von Variablen. PRVAR (  PRVAR) sucht den aktuellen Pfad für die Variablen, die Sie angegeben haben und druckt Namen und Inhalt jeder einzelnen Variablen im mehrzeiligen Format. PRVAR ruft einen Parameter vom Stack auf: Entweder einen einzelnen Namen oder eine Liste, die mehrere Namen enthält. (Mit PRVAR können Sie außerdem Sicherungsdaten drucken.)

Druck einer Textzeichenfolge

Sie können eine beliebige Folge von Zeichen drucken, indem Sie eine Zeichenkette eingeben, die die gewünschten Zeichen enthält und anschließend PR1 drücken. Der Drucker druckt die Zeichen ohne die Anführungszeichen und hält den Druckkopf am Ende der Druckzeile an. Der folgende Druck beginnt in der nächsten Zeile.

Druck von grafischen Objekten.

Wie bei anderen Daten, gibt es bei grafischen Objekten zwei Möglichkeiten des Druckes: Das grafische Objekt wird auf Ebene 1 gespeichert und läßt sich mit PR1 drucken. Die andere Möglichkeit besteht darin, das grafische Objekt in einer Variablen zu speichern, den Namen der Variablen einzugeben und PRVAR auszuführen.

Grafikdaten, die breiter als 166 Rasterpunktspalten sind, werden in 166 Spalten breiten Segmenten auf das Papier gedruckt; sie sind voneinander durch eine gestrichelte Linie getrennt. Zum Beispiel werden grafische Daten mit einer Breite von 350 Spalten in zwei Segmenten mit 166 Spalten und einem Segment mit 18 Spalten ausgegeben.

Zweizeiliger Druck

Um den zweizeiligen Druck (eine Leerzeile zwischen den Druckzeilen) zu wählen, müssen Sie Flag -37 (Zustandsmarke) setzen. Wenn Sie zum einzeiligen Druck zurückkehren wollen, ist Flag -37 zurück zu setzen.

Einstellen der Verzögerungszeit

Mit dem Befehl DELAY können Sie bestimmen, wie groß der zeitliche Abstand zwischen zwei Zeilen mit Informationen ist, die der HP 48 an den Infrarotdrucker HP 82240B überträgt. Der Befehl DELAY ruft eine reelle Zahl auf Ebene 1 auf, welche die Verzögerungszeit in Sekunden bestimmt. Wenn Sie keine Verzögerungszeit festlegen, wird diese automatisch auf 1,8 Sekunden eingestellt. Die Verzögerungszeit beträgt maximal 6,9 Sekunden.

Die Einstellung einer kleineren Verzögerungszeit kann nützlich sein, wenn der HP 48 mehrere Zeilen mit Informationen an Ihren Drucker überträgt (zum Beispiel beim Drucken eines Programms). Sie können die

Leistungsfähigkeit beim Drucken optimieren, indem Sie die Verzögerungszeit gerade etwas länger wählen als die Zeit, die der Druckkopf zum Drucken einer Zeile benötigt. Wenn Sie die Verzögerungszeit *kürzer* wählen als die Zeit zum Druck einer Zeile, können Ihnen Informationen verloren gehen. Wenn die Spannung der Batterien im Drucker geringer wird, wird der Druckkopf langsamer; in diesem Fall müssen Sie eine vorher bereits knapp eingestellte Verzögerung erhöhen, um den Verlust von Daten zu vermeiden. (Durch die Entladung der Batterien wird der Druckkopf allerdings nicht so langsam, daß er für den Druck einer Zeile länger als die eingestellte Standard-Verzögerungszeit von 1,8 Sekunden benötigt.)

Der Zeichensatz des HP 48

Die Tabelle in Anhang C enthält jedes einzelne Zeichen des HP 48 mit dem zugehörigen *Zeichencode*. Die meisten Zeichen in der Tabelle können direkt über die alphabetische Tastatur in das Anzeigefeld geschrieben werden. Wenn Sie zum Beispiel $\$$ darstellen wollen, müssen Sie    eingeben. (Die alphabetische Tastatur wird in Kapitel 2 vorgestellt.) Jedes Zeichen in dieser Tabelle läßt sich anzeigen, indem Sie den entsprechenden Zeichencode eingeben und anschließend den Befehl CHR ausführen. Die Syntax lautet *char# CHR*. Bestimmte Zeichen in der Tabelle in Anhang C sind *nicht* auf der alphabetischen Tastatur zu finden. Um eines dieser Zeichen im Anzeigefeld darzustellen, *müssen* Sie seinen Zeichencode eingeben und CHR ausführen.

Der Infrarotdrucker HP 82240B kann alle Zeichen aus dem Zeichensatz des HP 48 drucken.

Senden von Escape-Sequenzen und Steuercodes

Sie können verschiedene Betriebsarten des Druckers wählen, indem Sie *Escape-Sequenzen* an den Drucker senden. Eine Escape-Sequenz besteht aus dem Zeichen ESC (Escape) — Zeichen 27 —, dem zusätzliche Zeichen folgen. Empfängt der Drucker eine Escape-Sequenz, schaltet er in den gewählten Modus. Die Escape-Sequenz selbst wird nicht gedruckt.

Im allgemeinen sind in den Drucker-Handbüchern die Escape-Sequenzen und Steuercodes beschrieben, die der Drucker erkennt.

Erzeugen Sie die Escape-Sequenzen mit CHR und + und senden Sie sie mit PR1 an den Drucker.

Beispiel. Diese Zeichen senden Informationen an den Drucker HP 82240B, um den Modus "Unterstreichen" einzuschalten, unterstreichen die Zeichenfolge HELLO, und schalten danach den Modus "Unterstreichen" aus:

```
27 CHR 251 CHR + "HELLO" +  
27 CHR + 250 CHR + PR1
```

Speichern von Daten im Pufferspeicher des Druckers

Sie können eine beliebige Kombination aus Text, Grafik und Gegenständen in einer einzigen Zeile drucken, indem Sie die Daten im Puffer des Druckers akkumulieren (speichern).

Normalerweise schließt jeder Druckbefehl die Datenübertragung *automatisch* mit dem Befehl CR (*Druckkopf nach rechts*) ab, durch den der Drucker aufgefordert wird, einen Druckkopfrücklauf/Zeilenvorschub auszuführen. Der Drucker druckt dann die aktuellen Daten, die in seinem Pufferspeicher stehen, und hält den Druckkopf am rechten Ende der Druckzeile an.

Sie können die automatische Ausführung des Befehls CR sperren, indem Sie das Zeilenvorschub-Flag, das Flag -38 setzen. Die Daten der nächsten Druckbefehle werden im Puffer des Druckers gesammelt (zwischengespeichert) und nur gedruckt, wenn Sie den Befehl CR durch Tastendruck erzeugen. Wenn Flag -38 gesetzt ist, müssen Sie die folgenden drei Punkte beachten:

- Geben Sie den Befehl CR ( **PRINT**  CR ) ein, um die im Pufferspeicher gesammelten Daten zu drucken. (Es ist auch möglich, das Zeichen 4 oder das Zeichen 10 zu übertragen.)
- Drucken Sie die Daten im Puffer, bevor Sie mehr als 200 Zeichen gespeichert haben. Sonst läuft der Puffer über, und die folgenden Zeichen gehen verloren.
- Achten Sie darauf, daß der Drucker eine vollständige Zeile drucken kann, bevor Sie neue Daten übertragen. Für eine vollständige Zeile benötigt der Drucker ungefähr 1,8 Sekunden.

Setzen Sie das Flag -38 zurück, wenn Sie die normale Ausführung der Druckbefehle wiederherstellen wollen.

Drucken mit einem Infrarotdrucker HP 82240A

Sie können Ihren Rechner HP 48 zusammen mit einem Infrarotdrucker HP 82240A betreiben und dabei dieselben Druckbefehle wie bei einem HP 82240B verwenden. Der Zeichensatz im Infrarotdrucker HP 82240A entspricht allerdings nicht dem Zeichensatz des HP 48.

- 24 Zeichen in Zeichensatz des HP 48 sind im Infrarotdrucker HP 82240A nicht vorhanden. (In der Tabelle von Anhang C entsprechen diese Zeichen den Codes 129, 130, 143-157, 159, 166, 169, 172, 174, 184 und 185.) Anstelle dieser Zeichen druckt der HP 82240A ein ☒.
- Einige Zeichen in der erweiterten Zeichentabelle (Zeichencodes 128 bis 255) haben nicht den gleichen Zeichencode. Zum Beispiel hat das Zeichen « im HP 48 den Code 171 und den Code 146 im Infrarotdrucker HP 82240A. Wenn Sie den Befehl CHR dazu verwenden wollen, Zeichen des erweiterten Zeichensatzes mit einem Infrarotdrucker HP 82240A zu drucken, müssen Sie zuerst den Befehl OLDPRT ausführen. OLDPRT fügt der Variablen *PRTPAR* eine Zeichenfolge hinzu, die den Zeichencode der zu druckenden Bytes in die Codes in der Zeichentabelle des Infrarotdruckers HP 82240A umsetzt. (Wenn Sie eine Zeichenfolge drucken wollen, die grafische Daten enthält, darf der Befehl OLDPRT *nicht* wirksam sein.)

Wenn Sie OLDPRT verwendet haben, um mit einem Infrarotdrucker HP 82240A zu drucken und anschließend Ihre Daten mit einem Infrarotdrucker HP 82240B ausgeben wollen, müssen Sie zuerst die reservierte Variable *PRTPAR* löschen. (Vorher können Sie deren Inhalt in eine andere Variable kopieren, wenn Sie die Einstellungen für eine spätere Verwendung sichern wollen.) Damit werden die Druckparameter zurückgesetzt, so daß der Zeichensatz mit dem des HP 82240B übereinstimmt. (*PRTPAR* wird auf Seite 662 beschrieben.)

Drucken über den seriellen Anschluß

Über den seriellen Anschluß des HP 48 können Sie Daten zu einem seriellen Drucker übertragen. Wenn der Drucker an den HP 48 angeschlossen ist:

1. Setzen Sie das Flag für den Drucker, das Flag -34.

2. Kontrollieren Sie, ob das E/A-Flag, das Flag -33, freigegeben ist. (In der Standardeinstellung ist das Flag freigegeben.) *
3. Stellen Sie am HP 48 Übertragungsgeschwindigkeit (Baudrate), Parität und Übersetzungscode so ein, daß Sie für Ihren Drucker geeignet sind. Dazu können Sie das Menü I/O SETUP verwenden, das auf Seite 669 beschrieben wird.
4. Arbeitet Ihr Drucker im XON/XOFF-Quittungsbetrieb, müssen Sie *IOPAR* erstellen, um die Schrittsteuerung beim Senden der Daten auf einen Wert $\neq 0$ einzustellen. Die reservierte Variable *IOPAR* wird auf Seite 671 beschrieben.
5. Ist die Anzahl der Zeichen, die in eine Zeile Ihres Druckers passen, nicht gleich 80, müssen Sie *PRTPAR* so ändern, daß die richtige Zeichenzahl pro Zeile als drittes Element in seiner Liste steht. Die reservierte Variable *PRTPAR* wird im nächsten Abschnitt beschrieben.
6. Erwartet Ihr Drucker eine andere Sequenz zum Abschluß einer Zeile als Druckkopfrücklauf/Zeilenvorschub, müssen Sie *PRTPAR* so ändern, daß diese Sequenz als viertes Element in seiner Liste steht. Die reservierte Variable *PRTPAR* wird im folgenden Abschnitt beschrieben.

Sie können jeden einzelnen der in diesem Kapitel beschriebenen Befehle verwenden, wenn Sie mit einem seriellen Drucker arbeiten. Beachten Sie aber bitte folgendes:

- In der reservierten Variablen *PRTPAR* ist die maximale Zeilenlänge angegeben, die gedruckt wird (Diese Variable wird im nächsten Abschnitt behandelt.)
- Sie können keine grafischen Objekte drucken.

* Durch Setzen der beiden Marken -33 and -34 wird die serielle Datenübertragung über die Infrarotschnittstelle aktiviert. Wenn diese Flags gesetzt sind, ist der Druck mit einem Infrarotdrucker HP 82240B nicht möglich — der HP 82240B wird wahrscheinlich nur "Flecke" drucken.

Die Variable PRTPAR

Wenn Sie zum ersten Mal Informationen mit einem Befehl aus dem PRINT-Menü drucken, legt der HP 48 automatisch die Variable *PRTPAR* an. *PRTPAR* ist eine reservierte Variable, die in einer Liste alle Parameter enthält, die bestimmen, wie die Übertragung der Daten zwischen dem HP 48 und dem Drucker abläuft. Sie stehen in dieser Liste in der gleichen Reihenfolge, in der sie im folgenden beschrieben werden:

- Eine reelle Zahl, die die Verzögerungszeit in Sekunden bestimmt. Wenn Sie vor der Übertragung Ihrer Daten zum Drucker nicht den Befehl DELAY verwendet haben, wird die Verzögerungszeit in *PRTPAR* automatisch auf 1,8 Sekunden eingestellt.
- Eine Zeichenfolge, die die aktuell gültige Umsetzung des erweiterten Zeichensatzes des HP 48 darstellt. Die Zeichenfolge kann so viele Zeichen enthalten, wie Sie umsetzen wollen; dabei ist das erste Zeichen in der Zeichenfolge das neue Zeichen 128, das zweite ist das neue Zeichen 129, und so weiter. (Alle Zeichen außerhalb dieser Zeichenfolge werden nicht umgesetzt.) Wenn vor dem Senden Ihrer Daten nicht OLDPRT ausgeführt wurde, ist die Zeichenfolge leer. Wurde OLDPRT ausgeführt, enthält die Zeichenfolge die Codes der Zeichen für den Infrarotdrucker HP 82240A.
- Eine reelle Zahl, die die Zeilenlänge beim seriellen Druck durch die Anzahl der Zeichen festlegt. Dieser Parameter beeinflusst *nicht* das Drucken mit dem Infrarotdrucker. Die Standardeinstellung ist 80 Zeichen.
- Eine Zeichenfolge, die den Zeilenabschluß beim Drucken über den seriellen Anschluß festlegt. Dieser Parameter beeinflusst *nicht* das Drucken mit dem Infrarotdrucker. Als Standard ist *Druckkopfrücklauf/Zeilenvorschub* eingestellt (Steuerzeichen 13 and 10).

Sie können jeden Parameter in der Liste verändern. Die Verzögerungszeit läßt sich allerdings einfacher mit dem Befehl DELAY einstellen: Geben Sie eine entsprechende Zahl für die Verzögerungszeit ein (6,9 oder weniger) und führen Sie DELAY ( PRINT  NEXT DELAY) aus.

Datenübertragung vom und zum HP 48



Dieses Kapitel beschreibt:

- Die Datenübertragung von einem HP 48 zu einem zweiten HP 48 über den Infrarotanschluß.
- Die Datenübertragung zwischen dem HP 48 und einem Computer über den seriellen Anschluß. (Hierzu benötigen Sie eine entsprechende serielle Schnittstelle an Ihrem Computer. Nähere Informationen dazu erhalten Sie bei Ihrem HP-Händler.)
- Andere serielle E/A-Funktionen.

Der HP 48 verwendet das Protokoll *Kermit* für das Senden von Dateien, um Daten zu übertragen und um Übertragungsfehler zwischen zwei HP 48-Taschenrechnern oder zwischen einem HP 48 und einem Computer zu korrigieren. Das Protokoll Kermit wurde am *Center for Computing Activities* der Columbia-Universität entwickelt.

Die erforderlichen Computerbefehle zur Übertragung von Daten mit dem Kermit-Protokoll sind im HP 48 eingebaut. Deshalb können Sie Daten von einem HP 48 zu einem zweiten HP 48 übertragen, indem Sie einfach die beiden Infrarotanschlüsse einander gegenüberstellen und die entsprechenden Befehle geben. Die Befehle werden in diesem Kapitel beschrieben.

Um Daten von und zu einem Computer übertragen zu können, muß dieser mit einem Programm arbeiten, das das Kermit-Protokoll implementiert. Es ist außerdem ein Kabel erforderlich, das den HP 48 und den Computer miteinander verbindet. Einzelheiten zur Kabelverbindung werden weiter hinten in diesem Kapitel behandelt. (Für diese Funktion werden das Kermit-Protokoll und ein spezielles, serielles Kabel benötigt; beides gehört zum Lieferumfang einer nachrüstbaren, seriellen Schnittstelle, die Ihnen Ihr Hewlett-Packard-Händler liefert.) Wenn Sie sich ausführlicher mit dem Kermit-Protokoll beschäftigen wollen: zusätzliche Informationen finden Sie in einem Buch von Frank da Cruz, *KERMIT, A File Transfer Protocol*. *

Der HP 48 verfügt über zusätzliche, serielle E/A-Befehle für Daten, die nicht nach dem Kermit-Protokoll übertragen werden sollen. Diese Befehle sind für besondere E/A-Operationen vorgesehen — zum Beispiel, wenn Sie Daten direkt vom HP 48 zu einem Drucker mit serieller Schnittstelle übertragen wollen.

Arten von Daten, die Sie übertragen können

Eine Informationseinheit, die mit dem Kermit-Protokoll übertragen wird, heißt *Datei*. In der Welt des HP 48 ist unter einer Datei folgendes zu verstehen:

- Ein benanntes Objekt (Variable, ausgelagerte, usw.).
- Ein vollständiges Verzeichnis. Wenn Sie ein Verzeichnis übertragen, werden die Inhalte aller in diesem Verzeichnis vorhandenen Unterverzeichnisse ebenfalls übertragen.
- Alle Daten im Benutzerspeicher — alle Variablen, die Sie erzeugt haben, die Zuordnungen der Benutzertasten und der Terminkatalog.

In allen Fällen wird eine *Kopie* der Daten zum empfangenen Gerät übertragen und als Datei (Variable) im aktuellen Verzeichnis abgelegt.

* da Cruz, Frank. 1987. *KERMIT, A File Transfer Protocol*. Bedford, MA: Digital Press.

Wenn Sie ein Verzeichnis oder den gesamten Inhalt des Benutzerspeichers von einem HP 48 zu einem Computer übertragen, werden die Daten als eine einzige Datei gesendet, und Sie können nicht ohne weiteres auf den Inhalt der einzelnen Variablen in dieser Datei zugreifen. Aus diesem Grund ist zu empfehlen, das Verzeichnis hauptsächlich zur Archivierung an einen Computer zu senden.

Wenn Sie eine Datei übertragen wollen, um diese auf dem Zielrechner zu bearbeiten, (zum Beispiel, um ein Programm auf Ihrem Computer zu editieren), müssen Sie den Inhalt der einzelnen Variablen senden. Stellen Sie die Variablennamen in einer Liste zusammen und übertragen Sie die Daten mit dem Befehl SEND; Sie können dann auf die einzelnen Variablen zugreifen.

Wenn Sie ein Verzeichnis von einem HP 48 zu einem anderen übertragen, wird es als normales Verzeichnis im Zielrechner abgelegt. Das bedeutet, daß es genauso wie andere Verzeichnisse behandelt werden kann und daß alle seine Variablen frei verfügbar sind. Die Übertragung eines Verzeichnisses von einem HP 48 zu einem anderen ist eine geeignete Methode, eine Gruppe zusammengehörender Daten zu übertragen — zum Beispiel eine Gruppe von Programmen, Variablen, Druckerkonfigurationen usw. — die Sie gemeinsam auf dem Zielrechner HP 48 verarbeiten wollen.

Das E/A-Menü zur Ein- und Ausgabe von Daten

Die Befehle für das Kermit-Protokoll und die damit zusammenhängenden seriellen Operationen sind im E/A-Menü zur Ein- und Ausgabe von Daten enthalten. Die Befehle für die serielle Übertragung der Daten ohne Kermit-Protokoll werden am Ende des Kapitels behandelt.

Kermit-Befehle

Tasten	Programmierbarer Befehl	Beschreibung
  (Seiten 1 und 2):		
SEND	SEND	Sendet den Inhalt von einer oder mehreren Variablen an ein anderes Gerät. SEND benötigt dazu auf Ebene 1 den Variablennamen oder eine Liste von Namen { <i>Name₁ Name₂ . . .</i> } (Weitere Informationen hierzu finden Sie in dem Abschnitt unmittelbar nach dieser Tabelle.)
RECV	RECV	Fordert den HP 48 auf, auf den Empfang einer Variablen von einem anderen Gerät zu warten, das seine Daten ebenfalls nach dem Kermit-Protokoll überträgt.
SERVE	SERVER	Versetzt den HP 48 in den Kermit-Servermodus. (Diesen Befehl können Sie auch durch Betätigen von   erzeugen.)
KGET	KGET	Holt sich eine oder mehrere Variable von einem Server. KGET benötigt dazu auf Ebene 1 den Namen der gewünschten Variablen oder eine Liste von Namen { <i>Name₁ Name₂ . . .</i> }. (Weitere Informationen hierzu finden Sie in dem Abschnitt unmittelbar nach dieser Tabelle.)

Kermit-Befehle (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
FINIS	FINISH	Gibt den Kermit-Befehl FINISH an den Server, um den Servermodus zu beenden.
SETUP		Zeigt das Menü SETUP zur Einstellung der E/A-Parameter an.
RECN	RECN	Bewirkt dasselbe wie <code>RECV</code> , benötigt jedoch als Argument einen Namen. Die empfangene Datei wird unter diesem Namen gespeichert.
PKT	PKT	Bietet die Möglichkeit, ein "Paket" von Kermit-Befehlen an einen Server zu senden. Benötigt das Paketdatenfeld als Zeichenkette in Ebene 2 und den Pakettyp als Zeichenkette in Ebene 1. Zum Beispiel sendet die Buchstabenfolge "D" "G" PKT die Aufforderung, ein Verzeichnis aufzulisten.
KERR	KERRM	Liefert den Text der letzten Kermit-Fehlermeldung zurück.
OPENI	OPENIO	Öffnet den seriellen Anschluß, wobei die E/A-Parameter in <i>IOPAR</i> verwendet werden.
CLOSE	CLOSEIO	Schließt den seriellen Anschluß, löscht KERRM und löscht den Eingabepuffer.

Sie können SEND und KGET auch dazu verwenden, einer Variablen bei der Übertragung einen neuen Namen zu geben. Dazu muß die Namensliste eine Unterliste für diese Variable enthalten. Das erste Element in der Unterliste ist der aktuelle Variablenname und das zweite Element ist der neue Name. Wenn Sie zum Beispiel den Befehl SEND mit der Liste {*Name*₁ *Name*₂} *Name*₃ *Name*₄} als Argument anwenden, werden *Name*₃ und *Name*₄ unter ihrem eigenen Namen gesendet, während *Name*₁ mit dem neuen Namen *Name*₂ übertragen wird.

Lokaler Modus und Servermodus

Es gibt zwei Konfigurationen im Kermit-Protokoll, in denen Daten von einem HP 48 zu einem zweiten HP 48 oder zu einem Computer übertragen werden können:

- **Lokal/Lokal.** Beide Computer werden *lokal* über ihre eigenen Tastaturen bedient, und die Kermit-Befehle lassen sich von jedem der beiden Computer aus geben. Die Daten werden übertragen durch Eingabe des Befehls SEND über die Tastatur des Senders und durch Eingabe der Befehle RECV oder RECN über die Tastatur des Empfängers.
- **Lokal/Server.** Ein Computer wird *lokal* bedient und der andere Computer ist ein *Server*. Der Server wartet passiv auf Anweisungen oder auf Daten vom Sender. Ein Server
 - empfängt Daten, wenn ein Sender den Befehl SEND ausführt.
 - sendet Daten, wenn er den Befehl KGET empfängt.
 - beendet den Betrieb als Server, wenn er den Befehl FINISH empfängt.

Der Modus “Lokal/Server” ist sehr vorteilhaft, wenn Sie eine Anzahl von Variablen aus verschiedenen Dateiverzeichnissen übertragen wollen; das lokale Gerät kann mehrere Male Befehle zum “Senden” oder “Holen” ausgeben, ohne daß der Server bedient werden muß.

Einstellung der E/A-Parameter

Das Menü SETUP

Dücken von **SETUP** zeigt die zuletzt gewählten Einstellungen der E/A-Parameter und ein Menü, das es erlaubt, sie zu ändern. Wenn die angezeigten Einstellungen vom Stack oder anderen Informationen überschrieben werden, drücken Sie **← [REVIEW]**, um sie erneut anzuzeigen.

Das Menü SETUP

Tasten	Programmierbarer Befehl	Beschreibung
← [I/O] SETUP:		
IR/W		Schaltet um zwischen den Betriebsarten IR (Infrarot) und Wire (Serielle Leitung). Im IR-Modus wird das E/A-Ausgangssignal zum Infrarotanschluß geführt, im Wire-Modus zum seriellen Anschluß.
ASCII		Schaltet um zwischen den Übertragungsmodi "ASCII" und "Binär". (Einzelheiten hierzu finden Sie auf Seite 682).
BAUD	BAUD	Wählt die eine der Übertragungsgeschwindigkeiten 1200, 2400, 4800 und 9600 Baud. Die Standardeinstellung ist 9600 Baud.
PARIT	PARITY	Wählt schrittweise eine ungerade (1) oder eine gerade Parität (2), Mark (3), Space (4) oder keine (0) Parität. Bei der Standardeinstellung ist keine Parität gewählt.

Das Menü SETUP (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
CKSM	CKSM	Wählt schrittweise eine der Prüfsummenoptionen (Fehlererkennung). Eingestellt wird der Prüfsummentyp, der bei Auslösen des Befehls SEND erzeugt werden soll. Sie können wählen zwischen 1 (1-stellige arithmetische Prüfsumme), 2 (2-stellige arithmetische Prüfsumme) und 3 (3-stellige zyklische Redundanzprüfung, CRC). Die Standardeinstellung ist 3; für IR-Übertragungen sollten Sie immer Option 3 wählen.
TRAN	TRANSIO	Wählt schrittweise eine der Optionen zur Umsetzung der Zeichencodes. Sie können wählen zwischen 0 (keine Umsetzung), 1 (Umsetzen des Zeichens 10 in die Zeichen 13 und 10), 2 (Umsetzen der Zeichen 128 bis 159) oder 3 (Umsetzen der Zeichen 128 bis 255). Die Standardeinstellung ist 1. (Weitere Einzelheiten finden Sie auf Seite 679.)

Die Befehle BAUD, PARITY, CKSM und TRANSIO können in Programmen verwendet werden. Dabei muß dem Befehl die Zahl vorangestellt werden, die der gewählten Option entspricht.

Die Variable IOPAR

Die reservierte Variable *IOPAR* speichert die E/A-Parameter, die für den Aufbau einer Verbindung zur Datenübertragung zu einem Computer benötigt werden. *IOPAR* enthält eine Liste, die aus folgenden Elementen besteht:

{ *Baudrate Parität Empfangstaktsteuerung Sendetaktsteuerung Prüfsumme Übersetzungscode* }

IOPAR wird im Verzeichnis HOME angelegt, wenn Sie zum ersten Mal Daten übertragen oder den seriellen Anschluß öffnen (*OPENI*). Es wird immer dann automatisch aktualisiert, wenn Sie die Einstellungen mit Befehlen zur Wahl der E/A-Parameter aus dem Menü SETUP ändern.

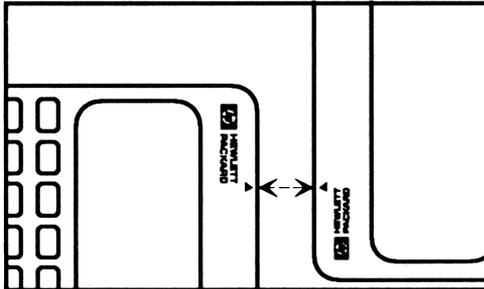
Einstellen der Parität. Ist die Paritätseinstellung positiv, wird sie für Senden und Empfangen verwendet. Ist sie negativ, wird sie nur beim Senden verwendet und die Parität wird beim Empfang nicht kontrolliert. Mit der Menütaste *PARIT* können Sie schrittweise nur positive Werte auswählen. Sie können aber eine negative Parität wählen, indem Sie die Zahl für die negative Parität in den Stack bringen, den Befehl *PARITY* eingeben und **ENTER** drücken. Eine andere Möglichkeit, das Paritätselement negativ zu machen, besteht darin, die Variable *IOPAR* zu ändern, die die zuletzt gewählten E/A-Parameter enthält.

Taktsteuerung beim Senden und Empfangen. Das Kermit-Protokoll arbeitet ohne Taktsteuerung beim Senden und Empfangen von Daten. Es ist jedoch möglich, sie bei anderen seriellen E/A-Übertragungen zu verwenden — zum Beispiel beim Drucken mit einem seriellen Drucker. Ein Wert ungleich Null für die Taktsteuerung beim Empfang veranlaßt den HP 48 dazu, ein XOFF-Signal zu senden, wenn sein Empfangspuffer voll wird und anschließend ein XON-Signal zu übertragen, wenn er weitere Daten aufnehmen kann. Ein Wert ungleich Null für die Taktsteuerung beim Senden bewirkt, daß der HP 48 den Sendevorgang abbricht, wenn er ein XOFF-Signal empfängt und auf ein XON-Signal wartet, um fortzufahren. Die Standardeinstellung dieser beiden *IOPAR*-Elemente ist 0, d.h. es werden keine XON/XOFF-Signale gesendet, und empfangene XON/XOFF-Signale werden nicht beachtet.

Übertragung von Daten zwischen zwei HP 48

Bevor Sie mit der Übertragung beginnen:

1. Rufen Sie im Sender das Verzeichnis auf, in dem sich die Variablen befinden. Wählen Sie im Menü SETUP zur Einstellung der E/A-Parameter den IR- und den Binär-Übertragungsmodus und stellen Sie CKSM auf 3.
2. Rufen Sie im Empfänger das Menü SETUP zur Wahl der E/A-Parameter auf, um den IR- Übertragungsmodus einzustellen. Rufen Sie anschließend das Verzeichnis auf, das die empfangenen Daten aufnehmen soll.
3. Richten Sie die ▲ Markierungen der Infrarotanschlüsse (neben dem Firmenzeichen von Hewlett-Packard direkt über der Anzeige) aufeinander aus. Die Computer dürfen nicht weiter als 5 cm voneinander entfernt aufgestellt werden.



Übertragung von Daten in der Konfiguration "Lokal/Lokal":

1. Am Empfänger haben Sie zwei Möglichkeiten:
 - Wenn Sie den Befehl RECV (\leftarrow I/O RECV) verwenden, dann wird die Variable unter dem Namen gespeichert, den der Sender vorgibt.
 - Wenn Sie den Namen der Variablen ändern wollen, geben Sie einen neuen Namen ein und geben dann den Befehl REC N (\leftarrow I/O NXT RECV). Nach dem Empfang wird das Objekt unter dem angegebenen Namen im Speicher abgelegt.

2. Geben Sie am Sender den Namen der Variablen oder des Verzeichnisses ein, die gesendet werden sollen, und geben Sie den Befehl SEND (  SEND). (Für mehrere Variablen in demselben Verzeichnis können Sie eine Liste dieser Variablen eingeben und sie mit SEND alle gemeinsam übertragen.)
3. Um weitere Variable oder Listen von Variablen zu senden, müssen beide vorherigen Schritte wiederholt werden.

Übertragung von Daten in der Konfiguration "Lokal/Server":

1. An dem HP 48, der als Server arbeiten soll, müssen Sie SERVER (  oder   SERVE) eingeben.
2. An dem anderen HP 48 sind je nach Funktion des Gerätes zwei Arten der Bedienung möglich:
 - Wenn Sie eine Datei zum Server übertragen wollen, müssen Sie den Namen der Variablen eingeben und SEND (  SEND) ausführen. (Wollen Sie die Variable mit einem anderen Namen senden oder verschiedene Variable aus demselben Verzeichnis übertragen, müssen Sie als Argument eine Liste entsprechend der Beschreibung auf Seite 668 verwenden.)
 - Wenn Sie eine Datei vom Server empfangen wollen, müssen Sie den Namen der Variablen und den Befehl KGET (  KGET) eingeben. (Wollen Sie die Variable nach dem Empfang unter einem anderen Namen speichern oder verschiedene Variable zusammen empfangen, müssen Sie als Argument eine Liste entsprechend der Beschreibung auf Seite 668 verwenden.)
3. Wiederholen Sie Schritt 2, wenn Sie weitere Variable oder Listen von Variablen senden wollen.
4. Geben Sie den Befehl FINISH (  FINIS) ein, um die Übertragung zu beenden.

Übertragung von Daten zwischen einem Computer und dem HP 48

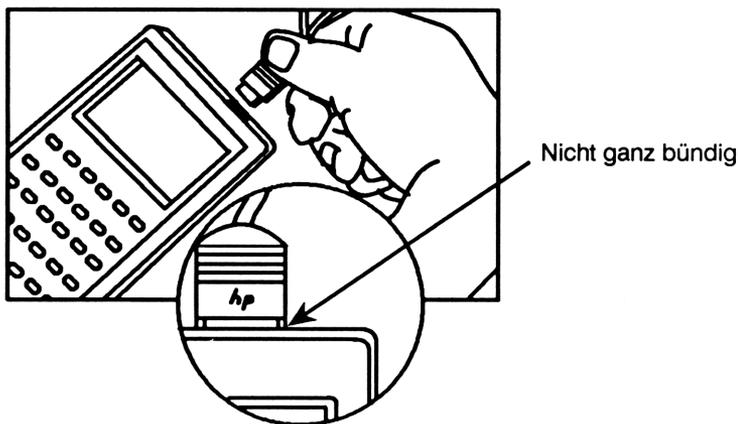
Es gibt viele Gründe, Informationen zwischen einem Computer und Ihrem HP 48 zu übertragen — vielleicht wollen Sie alle Daten im Anwenderspeicher Ihres Taschenrechners sichern; vielleicht wollen Sie ein Programm Ihres Taschenrechners auf dem Computer bearbeiten oder vielleicht wollen Sie auch ein Programm auf Ihrem Computer schreiben, das später auf Ihrem Taschenrechner laufen soll. In jedem Fall brauchen Sie als erstes eine Kabelverbindung.

Kabelverbindung

Bevor Sie Daten zwischen einem Computer und Ihrem Taschenrechner übertragen können, müssen Sie den HP 48 über das serielle Kabel, das zum Lieferumfang der nachrüstbaren, seriellen Schnittstelle für Ihren Computer gehört, an den Computer anschließen. (Wenn Sie Informationen dazu wünschen, mit welcher seriellen Schnittstelle Ihr Computer ausgestattet werden muß, oder wenn Sie keinen Nachrüstsatz für eine Schnittstelle haben, wenden Sie sich an Ihren HP-Händler.)

1. Verbinden Sie den Computerstecker des seriellen Kabels mit dem seriellen Anschluß am Computer. (Hierzu finden Sie entsprechende Hinweise im Handbuch Ihres Computers.)

2. Halten Sie den Taschenrechner mit der rechten Seite nach oben und nehmen Sie das Verbindungskabel so in die Hand, daß das HP-Firmenzeichen am Steckverbinder des Kabels nach oben zeigt. Schließen Sie das Kabel an Ihren Taschenrechner an. Dabei muß der Steckverbinder leicht einrasten.



Bitte beachten Sie: Wenn das Kabel richtig angeschlossen ist, steckt das Gehäuse des Steckverbinders nicht ganz im Gehäuse des Taschenrechners.

Übertragung von Daten

Bevor Sie mit der Übertragung beginnen:

1. Rufen Sie am HP 48 das Menü SETUP zur Wahl der E/A-Parameter auf (I/O) SETUP) und lesen Sie die Statusmeldung. Falls erforderlich:
 - Stellen Sie durch Betätigen der Taste den Übertragungsmodus "Wire" ein.
 - Wählen Sie den ASCII- oder den Binär-Modus für die Übertragung durch Drücken der Taste . (Auf Seite 682 finden Sie Hinweise zur Wahl des Übertragungsmodus.)
 - Stellen Sie für den HP 48 mit der Taste die gleiche Übertragungsgeschwindigkeit ein, mit der auch das Kermit-Programm arbeitet, das auf dem Computer läuft.

- Stellen Sie die Parität des HP 48 mit der Taste `PARIT` so ein, daß Sie mit der Einstellung des Kermit-Programms übereinstimmt, das auf dem Computer läuft.
 - Wählen Sie die Prüfsumme (`CKSM`)—Typ 1 läuft am schnellsten ab— und stellen Sie den Zeichenumsetzungscode ein (`TRAN`). (Seite 670 enthält Hinweise zur Wahl der Umsetzungs_codes.)
2. Rufen Sie am HP 48 und am Computer das Verzeichnis auf, in dem sich die zu sendenden Variablen (Dateien) befinden bzw. das Verzeichnis, in dem die empfangenen Variablen (Dateien) abgelegt werden sollen.
 3. Öffnen Sie den seriellen Anschluß des HP 48, indem Sie den Befehl `OPENIO` (`↵ I/O NXT OPENI`) verwenden. Dieser Schritt ist bei den meisten Datenverbindungen nicht erforderlich; damit vermeiden Sie aber Schwierigkeiten, die dadurch entstehen, daß bestimmte Geräte keine Daten zu einem geschlossenen Anschluß übertragen können.
 4. Lassen Sie auf dem Computer das Programm laufen, das das Kermit-Protokoll implementiert. Wenn Sie Ihre Daten im Binärmodus übertragen, so sollten Sie auch das Kermit-Programm auf dem Computer auf diese Betriebsart einstellen, vorausgesetzt, es verfügt über einen entsprechenden Befehl.

Übertragung von Daten in der Konfiguration “Lokal/Lokal”:

1. Geben Sie am Empfänger den Befehl “Empfangen”:
 - Ist der HP 48 der Empfänger, verwenden Sie den Befehl `RECV` (`↵ I/O RECV`), oder Sie geben einen Variablennamen ein und geben dann den Befehl `RECN` (`↵ I/O NXT RECN`).
 - Ist der Computer der Empfänger, geben Sie dort den Befehl, eine Datei zu empfangen.
2. Geben Sie am Sender den Befehl “Senden”:
 - Ist der HP 48 der Sender, geben Sie den Variablennamen (oder Variablenliste nach der Beschreibung auf Seite 668) ein und geben den Befehl `SEND` (`↵ I/O SEND`).
 - Ist der Computer der Sender, geben Sie dort den Befehl, eine Datei zu senden.
3. Wenn Sie weitere Variablen oder Variablenlisten übertragen wollen, müssen Sie die Schritte 1 und 2 wiederholen.

4. Wenn Sie nach der Übertragung den Befehl CLOSEIO ( I/O) (CLOSE) geben, werden die Batterien des Taschenrechners geschont.

Übertragung von Daten in der Konfiguration "Lokal/Server":

1. Wenn Ihr Computer als Server arbeiten soll, muß sicher sein, daß er den Befehl "Server" des Kermit-Protokolls ausführen kann.
2. Bereiten Sie das Gerät vor, das als Server eingesetzt werden soll:
 - Wenn der HP 48 als Server arbeiten soll, ist der Befehl SERVER ( I/O) oder ( I/O) SERV) auszuführen.
 - Soll der Computer als Server arbeiten, ist am Computer der Befehl auszuführen, der ihn zum Server macht.
3. Nehmen Sie an dem lokal gesteuerten Gerät folgende Einstellungen vor:
 - Geben Sie einen geeigneten Befehl zum "Senden", um eine Datei zum Server zu schicken. (SEND, falls der HP 48 Ihr Sender ist, eine Beschreibung finden Sie auf Seite 666.)
 - Geben Sie einen geeigneten Befehl zum "Holen", um eine Datei vom Server zu empfangen. (KGET, wenn der HP 48 Ihr Empfänger ist, eine Beschreibung finden Sie auf Seite 666.)
4. Wiederholen Sie Schritt 3, wenn Sie weitere Variablen übertragen wollen.
5. Geben Sie am lokal gesteuerten Gerät den Befehl, die Datenübertragung zu beenden. (Wenn der HP 48 als lokal gesteuertes Gerät arbeitet: drücken Sie  I/O) FINIS.)
6. Wenn Sie nach der Übertragung den Befehl CLOSEIO ( I/O) (CLOSE) am HP 48 geben, werden seine Batterien geschont.

Sichern aller Daten im Speicher des HP 48

Mit den Befehlen ARCHIVE und RESTORE haben Sie die Möglichkeit, alle im Taschenrechner gespeicherten Variablen, Zuweisungen von Benutzertasten und Termineinstellungen in Ihrem Computer zu sichern.

Um die gesamten im Benutzerspeicher abgelegten Daten zu sichern, sind folgende Bedienungsschritte erforderlich:

1. Nehmen Sie die Einstellungen entsprechend den Hinweisen im Abschnitt "Bevor Sie mit der Übertragung beginnen" auf Seite 675 vor.
2. Geben Sie das Objekt : IO: *Name* ein, wobei *Name* der Name der Datei ist, die die gesicherten Speicherdaten aufnehmen soll. Zum Beispiel wird durch : IO: AUG1 der Speicherinhalt in einer Datei mit dem Namen *AUG1* abgelegt.
3. Führen Sie den Kermit-Befehl RECEIVE am Computer aus.
4. Führen Sie den Befehl ARCHIVE ( MEMORY   ARCHI) aus, um Daten zum PC zu senden. (Unabhängig von der gewählten Übertragungsart — ASCII- oder Binärmodus — werden die Daten mit ARCHIVE binär übertragen.)

Kopieren der gesicherten Daten zurück in den HP 48:



Vorsicht

Wenden Sie den Befehl RESTORE mit Vorsicht an. Durch das Rückladen der gesicherten Daten in den Anwenderspeicher wird der aktuelle Speicherinhalt vollständig gelöscht und durch die Sicherungskopie ersetzt.

1. Nehmen Sie die Einstellungen entsprechend den Hinweisen im Abschnitt "Bevor Sie mit der Übertragung beginnen" auf Seite 675 vor.
2. Übertragen Sie die Datei aus dem Computer genau so zum HP 48, wie Sie jede andere Datei übertragen würden.
3. Bringen Sie den Namen der Datei in den Stack (zum Beispiel: 'AUG1 ') und drücken Sie   RCL]. Dadurch wird Backup HOMEDIR auf Ebene 1 geholt.
4. Geben Sie den Befehl RESTORE ( MEMORY   RESTO).

Wenn Sie die aktuellen Flag-Einstellungen beim Sichern der gesamten Speicherdaten mit archivieren wollen, müssen Sie RCLF ausführen und das Ergebnis in einer Variablen ablegen, bevor Sie den Speicherinhalt archivieren. Nach Archivieren und Rückladen des Speicherinhaltes können Sie den Inhalt der Variablen aufrufen und die Flag-Einstellungen durch Ausführen von STOF wieder aktivieren.

Umsetzen von Zeichen (TRANSIO)

Der Zeichensatz des HP 48 enthält bestimmte Zeichen, die von vielen Computerprogrammen nicht dargestellt werden können. Diese Zeichen lassen sich in zwei Gruppen unterteilen:

- Zeichen mit "Zeichennummern" im Bereich von 128 bis 159 können ohne besondere Programme zur Unterstützung des HP 48 nicht dargestellt werden.
- Zeichen mit den Zeichennummern im Bereich 160 bis 255 können von Computerprogrammen dargestellt werden, die mit dem Zeichensatz nach ISO 8859 arbeiten können.

Mit dem Umsetzungscode können Sie bestimmen, wie diese Zeichen vom HP 48 zu einem Computer übertragen werden. Sie wählen den Umsetzungscode mit dem Befehl TRANSIO. (TRAN in der Tabelle auf Seite 670 beschreibt die vier Umsetzungs-codes.)

Die folgende Tabelle zeigt, wie einige der Zeichen mit Nummern ab 127 konvertiert werden. Zeichen, die nicht in der Tabelle enthalten sind, werden in die Form `\xxx` umgewandelt, wobei `xxx` die dreistellige Zeichennummer darstellt.* Mit dieser Konvertierung ist es für Sie möglich, diese Zeichen mit dem Editor Ihres Computers einzugeben und darzustellen.

* Sie können diese Konvertierung auch auf Zeichen in der Tabelle und auf Zeichen von 0 bis 127 anwenden, was die Editierung in Steuerzeichen oder in einer Escapesequenz auf Ihrem Computer erleichtert. Der HP 48 erzeugt keine xxx-Sequenzen, kann sie aber erkennen.

Umsetzung von E/A-Zeichen.

Zch. Nummer	HP 48 Zch.	PC Zch.	Zch. Nummer	HP 48 Zch.	PC Zch.
128	\sphericalangle	\<)	147	ϵ	\Ge
129	\bar{x}	\x-	148	η	\Gn
130	∇	\.V	149	θ	\Gh
131	$\sqrt{\quad}$	\v/	150	λ	\Gl
132	\int	\.S	151	ρ	\Gr
133	Σ	\GS	152	σ	\Gs
134	\blacktriangleright	\ >	153	τ	\Gt
135	π	\pi	154	ω	\Gw
136	∂	\.d	155	Δ	\GD
137	\leq	\<=	156	Π	\PI
138	\geq	\>=	157	Ω	\GW
139	\neq	\=/	158	■	\[]
140	α	\Ga	159	∞	\oo
141	\rightarrow	\->	171	«	\<<
142	\leftarrow	\<-	176	\circ	\^o
143	\downarrow	\ v	181	μ	\Gm
144	\uparrow	\ ^	187	»	\>>
145	γ	\Gg	215	\times	\.x
146	δ	\Gd	216	\emptyset	\O/
			247	\div	\:-

Um Doppeldeutigkeiten bei der Umsetzung und der Rückumsetzung zu vermeiden:

- Wenn Daten vom HP 48 mit einem Umsetzungscode von 2 oder 3 übertragen werden, wird jedes Zeichen \backslash durch $\backslash\backslash$ ersetzt. Zum Beispiel wird $A\backslash\rightarrow B$ in $A\backslash\backslash\rightarrow B$ umgewandelt. Dadurch vermeiden Sie die Rückumsetzung in die Form $A\rightarrow B$, wenn die Daten zum HP 48 zurück übertragen werden.

- Wenn Daten *zum* HP 48 mit einem Umsetzungscode von 2 oder 3 übertragen werden, bleiben Zeichenfolgen, die mit \ beginnen, unverändert, wenn nicht eine der folgenden Bedingungen zutrifft:
 - Sie stimmen mit einer Sequenz in der Tabelle überein.
 - Beim Umsetzungscode 2 folgen dem \ drei Dezimalstellen im Bereich von 000 bis 159.
 - Beim Umsetzungscode 3 folgen dem \ drei Dezimalstellen im Bereich von 000 bis 255.

Zum Beispiel werden \G₃ and \215 in α und × umgewandelt; aber \G× und \267 werden nicht umgesetzt.

Mehr zu den Dateinamen

Es gibt Unterschiede zwischen den Konventionen für die Benennung von Dateien bei Computern und für die Bezeichnung von Variablen beim HP 48. Wird eine Datei von einem Computer zum HP 48 übertragen, können folgende Probleme auftreten, deren Ursache im gewählten Namen für die Computerdatei liegt:

- Der Dateiname enthält Zeichen, die nicht in einem Variablennamen enthalten sein dürfen — zum Beispiel: AE# oder {ABC}. In diesem Fall beendet der HP 48 die Datenübertragung und sendet eine Fehlermeldung an den Computer.
- Der Dateiname entspricht einem programmierten Befehl — zum Beispiel: SIN oder DUP. In diesem Fall, hängt der HP 48 dem Namen eine Zahl als Erweiterung an — zum Beispiel: SIN. 1.
- Der Name stimmt mit einem Variablennamen im aktuellen Verzeichnis überein. In diesem Fall wird dem Namen eine Zahl als Erweiterung angehängt, um zu vermeiden, daß Ihre Variable überschrieben wird. (Ist allerdings das Flag -36 gesetzt, wird die Variable überschrieben.)

Es ist auch möglich, daß eine Datei des HP 48 einen Namen hat, der nicht den Anforderungen des Computerprogramms an einen Dateinamen entspricht. Das Senden einer solchen Datei kann zu einem Übertragungsfehler führen.

Kontrollieren Sie vor einer Übertragung, ob die Dateinamen mit den Anforderungen des empfangenden Systems vereinbar sind. Ändern Sie die Namen entsprechend, wenn sie nicht kompatibel sind.

Fehler

Nach Ausführung des Befehls KERRM ( I/O  KERR) wird der Text des letzten Kermit-Fehlerpaketes angezeigt.

Übertragung im ASCII-Modus und im Binär-Modus

Beim HP 48 stehen im Kermit-Protokoll zwei Übertragungsarten zur Verfügung — der ASCII-Modus und der Binärmodus. Die schnellste Datenübertragung von einem HP 48 zu einem zweiten HP 48 bietet im allgemeinen der Binärmodus, während die Daten zwischen dem HP 48 und einem Computer am schnellsten im ASCII-Modus übertragen werden.

Ein HP 48 als Datenempfänger behandelt alle Dateien als ASCII-Dateien, wenn sie nicht die speziell für Binärdateien des HP 48 erzeugte Codierung tragen. Bei Dateien mit dieser Kodierung schaltet der Taschenrechner automatisch in den Binärempfangmodus.

ASCII-Modus. Sie *müssen* Ihre Daten im ASCII-Modus übertragen, wenn Sie eine HP 48-Datei auf einem Computer anzeigen, bearbeiten oder drucken wollen.

Wenn Daten im ASCII-Modus vom HP 48 zu einem Computer übertragen werden, ist folgendes zu beachten:

- Die Daten werden aus ihrem internen HP 48-Format in eine Folge von Zeichen umgewandelt.
- Ist der Umsetzungscode auf 1, 2 oder 3 eingestellt worden, werden alle Zeilenvorschub-(LF)-Zeichen in Druckkopfrücklauf-/Zeilenvorschubsequenzen (CR/LF) umgewandelt.
- Ist der Übersetzungscode auf 2 oder 3 eingestellt worden, werden einige oder alle Zeichen mit Zeichennummern größer als 127 in darstellbare Zeichenfolgen umgesetzt.
- Die Zeichenfolge `⌘HP: Modi` ; wird an den Anfang der Daten gesetzt, wobei *Modi* eine Reihe von Zeichen darstellt, die bestimmte Moduseinstellungen des Taschenrechners — die Einstellungen für Übersetzung, Winkel und Dezimalzeichen — beschreiben die gültig waren, als die Daten übertragen wurden. Wenn diese Sequenz übertragen wird, müssen Sie die entsprechenden Einstellungen am HP 48 beim Rücksenden der Daten nicht mehr vornehmen.

Wenn der HP 48 Daten im ASCII-Modus empfängt; geschieht folgendes:

- Die Daten werden in das interne Format des HP 48 übersetzt.
- Ist der Umsetzungscode auf 1, 2 oder 3 eingestellt worden, werden alle CR/LF-Zeichen in LF-Zeichen konvertiert.
- Damit der empfangende Taschenrechner die vom Computer gesendeten Daten fehlerfrei rekonstruieren kann, wird er für die Dauer der Übertragung auf die am Anfang der Daten angegebenen Modi eingestellt. Existieren für einen Modus keine Angaben, so verwendet der empfangende Taschenrechner die zuletzt gültige Einstellung.

Wenn Sie mit Ihrem Computer Daten erstellt haben (zum Beispiel ein Programm) oder die ursprünglich von Ihrem Taschenrechner stammenden Daten wesentlich verändert haben, kann es sein, daß Sie den Daten die Zeichen `%%HP: Modi` ; voranstellen müssen. Dabei steht *Modi* für eine Reihe von Zeichen — T(), A(), und/oder F() — die den Umsetzungscode, den Winkelmodus und/oder das Dezimalzeichen darstellen. In den Klammern stehen die Zeichen, die Sie wählen:

- Bei T (Umsetzungscode) kann stehen: 0 (keine Umsetzung), 1 (Übersetze CR/LF in LF und umgekehrt), 2 (Übersetze CR/LF-Zeichen und Zeichennummern 128 bis 159) oder 3 (Übersetze CR/LF-Zeichen und Zeichennummern 128 bis 255).
- Bei A (Winkelmodus) kann stehen : D (Grad), R (Radianten), or G (Gon). Enthalten die Daten einen Winkel in Grad, Radianten oder Gon, ist der Winkelmodus mit A(D), A(R) bzw. A(G) zu bezeichnen.
- Bei F (Dezimalzeichen) kann stehen: (Punkt) oder , (Komma). Wenn es sich von dem am Taschenrechner eingestellten Dezimalzeichen unterscheidet, muß das in den gesendeten Daten verwendete Zeichen durch F(.) bzw. F(,) gekennzeichnet werden.

Zum Beispiel bewirkt die am Anfang der gesendeten Daten stehende Zeichenfolge: `%%HP:A(D)`, daß der Winkelmodus bei der Übertragung auf Grad eingestellt wird; `%%HP:T(2)A(G)F(,)` wählt den Umsetzungscode 2, stellt den Winkelmodus auf Gon und das Dezimalzeichen auf Komma.

Der Umsetzungscode T(1) ist der übliche Fall (und daher auch die Standardeinstellung des Systems). Sie sollten T(2) oder T(3) nur dann verwenden, wenn Zeichen in den jeweiligen Bereichen entsprechend der Tabelle auf Seite 670 übersetzt werden sollen. Der Code T(0) ist nur auf Zeichenfolgen oder Objekte mit Zeichenfolgen anzuwenden, wenn die Zeichenfolgen Binärdaten enthalten.

Binärmodus. Im Binärmodus werden keine Zeichen konvertiert. Deshalb kann der Computer Daten, die er vom HP 48 empfangen hat, nicht anzeigen. Wenn Daten allerdings nur zur Sicherung übertragen werden, ist der Binärmodus vorzuziehen. Die Datenübertragung erfolgt schneller, weil dieser Modus keine so aufwendige Verarbeitung der Daten erfordert.

Der HP 48 wählt automatisch den Binärmodus beim Senden von Bibliotheken, bei der Übertragung von ausgelagerten Objekten oder bei der Archivierung aller Daten des Anwenderspeichers.

Senden von Befehlen an einen Server (PKT)

Der Befehl PKT ( I/O  PKT) bietet die Möglichkeit, auch andere Daten als HP 48-Objekte an einen Server zu senden oder von ihm zu empfangen. Er ist besonders nützlich beim Senden von Kermit-Befehlen — zum Beispiel: Directory (D) oder Erase (E).

Der Befehl PKT benötigt als Argumente zwei Zeichenketten auf dem Stack — das Datenfeld des Pakets in Ebene 2 und den Pakettyp in Ebene 1. Zum Beispiel wird durch die Sequenz "D" "G" PKT die Auflistung eines Inhaltsverzeichnisses angefordert.

Ein Server reagiert auf eine der folgenden Arten auf den Befehl PKT:

- Er schickt eine Bestätigungsmeldung, die auf Stack 1 abgelegt wird.
- Er schickt ein Fehlerpaket. Der HP 48 zeigt den Inhalt des Fehlerpaketes kurz an. Es kann durch den Befehl KERRM ( I/O  KERR) abgerufen werden.

Befehle für die serielle Ein- und Ausgabe



Vorsicht

Wenn Sie Daten von oder zu einem HP 48 mit einer Übertragungsrate von 9600 Baud senden wollen, sollten Sie vorher sicherstellen, daß die Uhr im Anzeigefeld nicht zu sehen ist. Wird die Uhr angezeigt, so kann es sein, daß eine Datenübertragung unterbrochen wird oder die zu übertragenden Daten verfälscht werden. Die Uhr in der Anzeige ist in Kapitel 24 "Zeit, Meldungen und Datenarithmetik" beschrieben.

Serielle E/A-Befehle

Tasten	Programmierbarer Befehl	Beschreibung
  (page 3):		
	XMIT	Sendet eine Zeichenkette, die in Ebene 1 steht, ohne Kermit-Protokoll. Nach der Übertragung der gesamten Zeichenkette wird eine 1 in Ebene 1 abgelegt. Konnte die Zeichenkette nicht vollständig übertragen werden, so wird eine 0 in Ebene 1 abgelegt, und der nicht gesendete Teil der eingegebenen Zeichenkette wird nach Ebene 2 geschoben. Mit dem Befehl ERRM kann die Fehlermeldung angezeigt werden.

Serielle E/A-Befehle (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
SRECV	SRECV	<p>Empfängt x Zeichen (Parameter x wird von Ebene 1 geholt). Die Zeichen werden als Zeichenkette in Ebene 2 abgelegt; dazu eine 1 (fehlerfreier Empfang) oder eine 0 (kein fehlerfreier Empfang) in Ebene 1. Enthält der Eingangspuffer weniger als x Zeichen, wartet der HP 48 eine bestimmte Zeit ab. Die Zeitdauer in Sekunden wird durch den Befehl STIME festgelegt (die Standardeinstellung ist 10 Sekunden). (Wird für Argument SRECV die Zahl verwendet, die der Befehl BUFLN in Ebene 2 schreibt, so entsteht keine Wartezeit, weil x dann genau mit der Anzahl der Zeichen im Eingangspuffer übereinstimmt. BUFLN ist in der Tabelle auf Seite 687 beschrieben.) Bei einem fehlerhaften Empfang der Daten wird nach Ausführen von ERRM eine entsprechende Fehlermeldung angezeigt.</p>
STIME	STIME	<p>Stellt ein, nach welcher Wartezeit beim seriellen Senden/Empfangen die Verbindung unterbrochen werden soll (das Argument x wird aus der Ebene 1 geholt). Der Wert für x kann zwischen 0 und 25,4 Sekunden liegen. Bei der Einstellung auf 0 wird nicht unterbrochen.</p>

Serielle E/A-Befehle (Fortsetzung)

Tasten	Programmierbarer Befehl	Beschreibung
SBRK	SBRK	Schreibt ein BREAK an den seriellen Anschluß.
BUFLE	BUFLEN	Schreibt die Anzahl der Zeichen im Eingangspuffer des HP 48 nach Ebene 2; dazu eine 1 (kein Synchronisierfehler oder UART-Überlauf) oder eine 0 (Synchronisierfehler oder UART-Überlauf) nach Ebene 1. Bei einer 0 entspricht die in Ebene 2 stehende Anzahl der Zeichen dem Teil der Daten, der vor Auftreten des Fehlers empfangen wurde. Deshalb kann mit Hilfe dieser Zahl festgestellt werden, wo der Fehler aufgetreten ist.



Hinweis

Obwohl XMIT, SRECV und BUFLN die Send- und Empfangsmechanismen steuern, wird die Vollständigkeit der Daten nicht überprüft. Um zu garantieren, daß die empfangenen mit den gesendeten Daten identisch sind, fügt man eine Prüfsumme an die gesendeten Daten und vergleicht sie mit der beim Empfang gebildeten Prüfsumme.

XMIT, SRECV und SBRK öffnen automatisch den IR-/seriellen Anschluß, wobei sie die aktuellen Werte der ersten vier IOPAR-Parameter (Übertragungsrates, Parität, Taktsteuerung beim Empfang und Taktsteuerung beim Senden) sowie die zuletzt gewählte Einstellung auf Infrarot oder Leitung verwenden (mit IR/W im Menü I/O SETUP änderbar).

Verwendung von Einsteckkarten und Bibliotheken



In diesem Kapitel werden behandelt:

- Die Arten der Speicher- und sonstigen Einsteckkarten
- Einbau und Ausbau von Einsteckkarten
- Verwendung von RAM-Karten zur Erweiterung des Benutzerspeichers oder zur Sicherung von Daten.
- Verwendung von Anwender-Steckkarten und Bibliotheken

Speicherarten

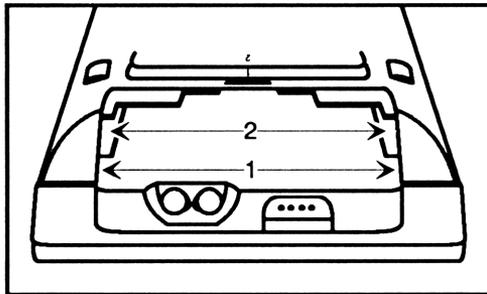
Einsteckkarten erweitern den Speicherbereich des HP 48. Der HP 48 arbeitet mit zwei unterschiedlichen Speichern:

- *Festwertspeicher* oder ROM (Read-Only-Memory) ist der Speicher, dessen Inhalt nicht verändert werden kann. Das interne ROM des HP 48 hat eine Kapazität von 256 KByte und enthält den Befehlssatz des Taschenrechners. Der ROM-Bereich läßt sich durch den Einbau von Steckkarten mit Bibliotheken erweitern.

- *Schreib-Lese-Speicher* oder RAM (Random Access Memory) ist der Speicher, dessen Inhalt verändert werden kann. Sie können Daten im RAM speichern, seinen Inhalt verändern und Daten löschen. Das interne RAM des HP 48 hat eine Kapazität von 32 KByte. Sie können den RAM-Bereich durch den Einbau von RAM-Einsteckkarten erweitern.

Einbau und Ausbau von Einsteckkarten

Der HP 48 hat zwei *Einschübe* für die Installation von Einsteckkarten; diese werden mit Port 1 und Port 2 bezeichnet. Port 1 befindet sich näher an der Vorderseite des Taschenrechners (weiter innen); Port 2 ist näher an der Rückseite (weiter außen). Die Karten können an beide Ports angeschlossen werden.



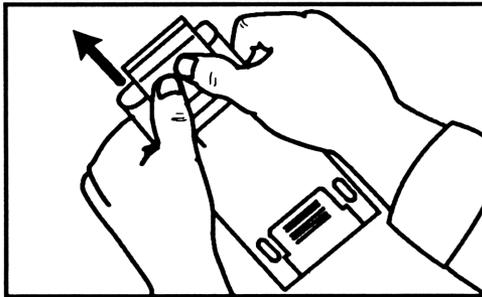
Vorsicht

Der Taschenrechner muß beim Ein- oder Ausbau der Einsteckkarten ausgeschaltet sein, da sonst der Inhalt des Benutzerspeichers gelöscht werden kann.

Der Ein- oder Ausbau einer Karte bewirkt eine Systemunterbrechung, was bedeutet, daß der Inhalt des Stacks verlorengeht.

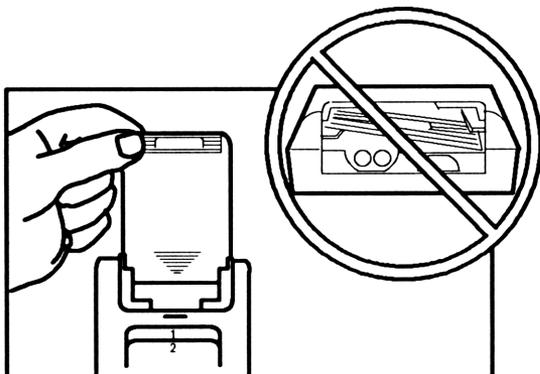
Bei der Installation einer Einsteckkarte gehen Sie bitte folgendermaßen vor:

1. Wenn Sie eine neue RAM-Karte installieren, müssen Sie zuerst ihre Batterie einsetzen (Einzelheiten finden Sie im Abschnitt “Einbau der Batterie in eine neue RAM-Karte” auf Seite 693) und die erforderliche Stellung des Schreibschutzschalters wählen (Einzelheiten finden Sie im Abschnitt “Einstellen des Schreibschutzschalters” auf Seite 696).
2. Schalten Sie den Taschenrechner aus. *Drücken Sie die Taste **ON** erst wieder, wenn Sie die Installation der RAM-Karte beendet haben.*
3. Entfernen Sie die Abdeckung der Ports am Taschenrechner, indem Sie auf die Grifffläche drücken und die Abdeckung in die dargestellte Richtung schieben. Nach dem Entfernen der Abdeckung sehen Sie die beiden Ports.



4. Wählen Sie ein freies Port für die Karte — Sie können beide Ports verwenden.

5. Halten Sie die Einsteckkarte wie dargestellt fest. Der dreieckige Pfeil muß nach unten in Richtung auf den Taschenrechner zeigen. Achten Sie darauf, daß die Karte richtig in eine Portöffnung hineingeschoben und dabei nicht verkantet wird.



6. Schieben Sie die Karte fest bis zum Anschlag in das Port. Beim ersten fühlbaren Widerstand hat die Karte noch etwa 6 mm bis sie anstößt.
7. Zum Einbau einer weiteren Karte wiederholen Sie die Schritte 4 bis 6.
8. Schieben Sie die Abdeckung wieder in ihre Führung, bis sie einrastet.
9. Wenn Sie eine RAM-Karte einsetzen, müssen Sie entscheiden, wie Sie sie verwenden wollen (Beachten Sie dazu Seite 696):
- Wollen Sie die RAM-Karte zur Erweiterung des Benutzerspeichers einsetzen, muß der Befehl MERGE entsprechend der Beschreibung auf Seite 698 ausgeführt werden.
 - Wollen Sie die RAM-Karte als unabhängigen Speicher einsetzen, muß der Befehl MERGE entsprechend der Beschreibung auf Seite 698 und anschließend der Befehl FREE (Seite 704) ausgeführt werden.

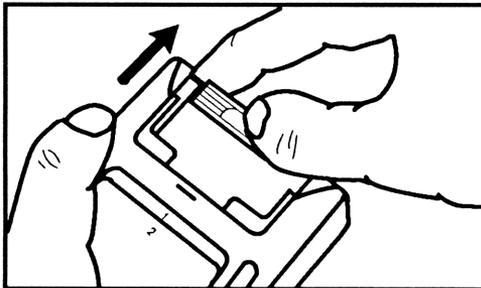
Beim Ausbau einer Einsteckkarte gehen Sie bitte folgendermaßen vor:



Vorsicht

Wenn die Einsteckkarte, die Sie ausbauen wollen, eine RAM-Karte ist, die einen Teil des Benutzerspeichers enthält, müssen Sie diesen Speicher vor dem Herausnehmen der Karte *freigeben*. Sonst ist es möglich, daß im Benutzerspeicher abgelegte Daten verloren gehen. Weitere Hinweise finden Sie im Abschnitt "Freigeben des eingebundenen Speichers" auf Seite 704.

1. Schalten Sie den Taschenrechner aus. *Drücken Sie die Taste **ON** erst wieder, wenn Sie den Ausbau der RAM-Karte beendet haben.*
2. Entfernen Sie die Portabdeckung.
3. Um eine Karte aus dem Taschenrechner herauszunehmen, müssen Sie - wie unten dargestellt - auf die Grifffläche drücken und die Karte aus dem Port herauschieben.



4. Schieben Sie die Portabdeckung wieder ein.

RAM-Karten

RAM-Karten ermöglichen die Erweiterung des Schreib-Lese-Speichers (RAM) in Ihrem HP 48. Jede Karte arbeitet mit einer eigenen Batterie, so daß der Speicherinhalt erhalten bleibt, wenn der Taschenrechner ausgeschaltet oder wenn die Karte ordnungsgemäß aus dem Taschenrechner herausgenommen wurde.

RAM-Karten ermöglichen:

- Erweiterung des Anwenderspeichers.
- Sicherung oder Geheimhaltung wichtiger Daten.
- Austausch von Daten zwischen zwei HP 48-Taschenrechnern.
- Speicherung von neu entwickelten Anwendungsprogrammen, die später möglicherweise in Festwertspeichern (ROM) abgelegt werden.

Der Abschnitt “Anwendungen für RAM-Karten” auf Seite 697 behandelt diese Themen.

Vorbereiten der Karte für den Einbau

Einbau der Batterie in eine neue RAM-Karte. Bevor eine neue RAM-Karte installiert werden kann, ist die mitgelieferte Batterie in die Karte einzubauen.



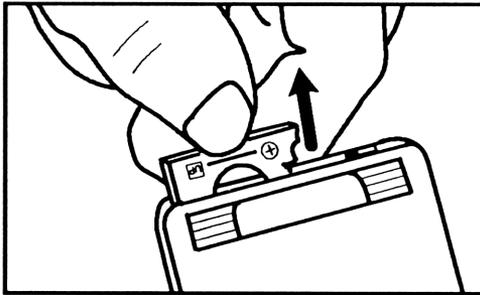
Vorsicht

Gehen Sie nicht nach dieser Beschreibung vor, wenn Sie die Batterie einer RAM-Karte *auswechseln* wollen — der Speicherinhalt der RAM-Karte kann verloren gehen. Anhang A enthält auf Seite 720 Hinweise zum

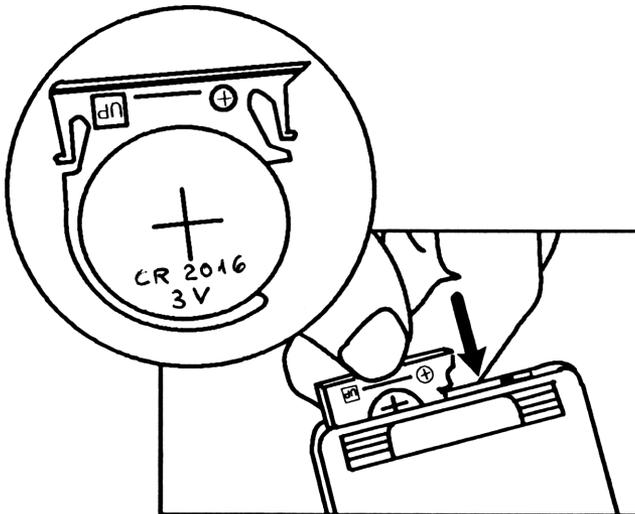
Auswechseln von RAM-Kartenbatterien.

Beim Einsetzen der Batterie in eine neue RAM-Karte ist wie folgt vorzugehen:

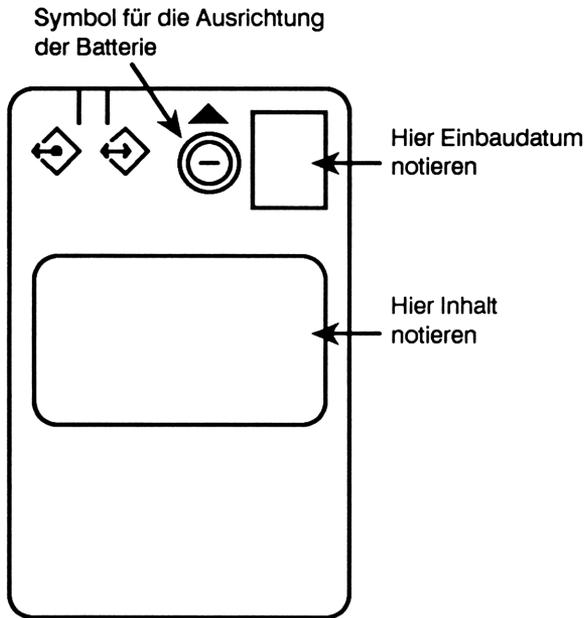
1. Nehmen Sie die Batteriehalterung aus der Karte heraus, indem Sie diese in die dargestellte Richtung ziehen. Zur Erleichterung hat die Halterung eine schmale Rille für Ihren Daumnagel oder für einen kleinen Schraubenzieher.



2. Die geriffelte Seite der Batteriehalterung ist mit dem Symbol + und dem Wort UP markiert. Setzen Sie die Batterie mit dem Pluspol nach oben in die Halterung und schieben Sie die Halterung in die Karte hinein.



3. Schreiben Sie das Datum des Batterieeinbaus mit einem wasserfesten Markierstift auf die Karte. So können Sie jederzeit feststellen, wann die Batterie ausgewechselt werden muß.



4. Stellen Sie im Taschenrechner einen Termin etwa 1 Jahr nach dem Datum der Installation ein. Dieser Termin soll Sie daran erinnern, die Batterie rechtzeitig auszuwechseln. (Die Batterie sollte je nach Häufigkeit der Benutzung zwischen 1 und 3 Jahren halten. Wenn die Batterie ausgewechselt werden muß, wird das durch eine Meldung angezeigt, *falls die Karte im Taschenrechner eingesetzt ist*. Den Termin sollten Sie für den Fall einstellen, daß die Karte nicht im Taschenrechner installiert ist, wenn die Batterie ausgewechselt werden muß.) Die Einstellung von Terminen ist in Kapitel 24 beschrieben, und Hinweise zum Auswechseln der RAM-Kartenbatterie finden Sie in Anhang A.

Einstellung des Schreibschuttschalters. Der Schreibschuttschalter ermöglicht den Schutz der Daten auf der RAM-Karte gegen unbeabsichtigtes Überschreiben oder Löschen. Der Schalter hat zwei Stellungen:

- **Nur-Lesen.** Der Inhalt der RAM-Karte kann nur gelesen, aber nicht verändert oder gelöscht werden.
- **Lesen/Schreiben.** Sie können Informationen auf die RAM-Karte schreiben und ihren Inhalt löschen.

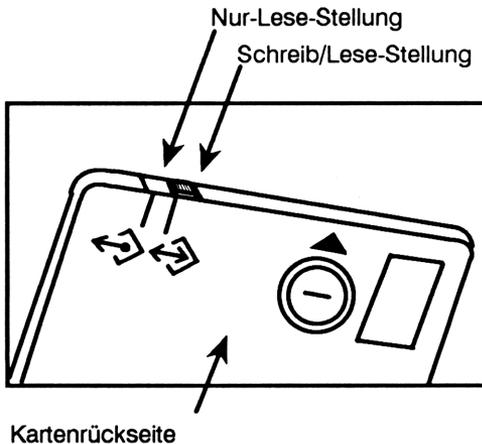


Vorsicht

Bitte beachten Sie folgende Hinweise, wenn Sie den Verlust von Daten im Benutzerspeicher vermeiden wollen:

- Schalten Sie immer erst den Taschenrechner aus, bevor Sie die Schalterstellung des Schreibschuttschalters an einer eingebauten Karte ändern.
- Verwenden Sie den Schreibschutz nicht bei einer RAM-Karte, die einen Teil des Benutzerspeichers enthält; der Speicher muß vorher freigegeben werden (Sehen Sie dazu Seite 704).

Sie können den Schreibschuttschalter auch an einer eingebauten Karte verstellen; die Schalterbezeichnungen sind allerdings nicht sichtbar.



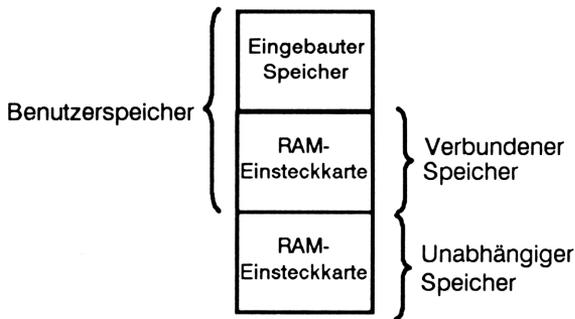
Anwendungen für RAM-Karten

Es gibt zwei Möglichkeiten, eine RAM-Karte zu verwenden:

- Der Speicher kann in den internen Speicher *eingebaut* werden. Dadurch können Sie den verfügbaren Bereich des Benutzerspeichers bis auf 288 KByte erweitern und haben auf diese Weise die Möglichkeit, mehr Variablen und Inhaltsverzeichnisse anzulegen, Objekte auf den Stack zu legen, usw.
- Sie können einen Speicherbereich anlegen, der vom Benutzerspeicher *unabhängig* ist und dort wichtige Daten sichern. Sie können individuelle Objekte oder ganze Inhaltsverzeichnisse in fast derselben Weise auf eine RAM-Karte kopieren, wie Sie Computerdateien auf einer Diskette sichern würden. Nachdem Sie die Daten kopiert haben, können Sie die Karte herausnehmen und an einem sicheren Ort aufbewahren. Sie können natürlich auch in einem anderen HP 48 installieren und auf diese Weise eine Art von Datenübertragung durchführen.

Sie können eine oder zwei RAM-Karten installieren, und Sie können die Karten einzeln oder zusammen für jede der beiden Anwendungsmöglichkeiten einsetzen. Sie können eine einzelne Karte allerdings nicht gleichzeitig als Teil des Benutzerspeichers und als unabhängigen Speicher einsetzen.

Das folgende Bild zeigt ein System, das aus zwei RAM-Karten besteht — eine Karte enthält einen Teil des Benutzerspeichers, und die andere enthält einen unabhängigen Speicher.

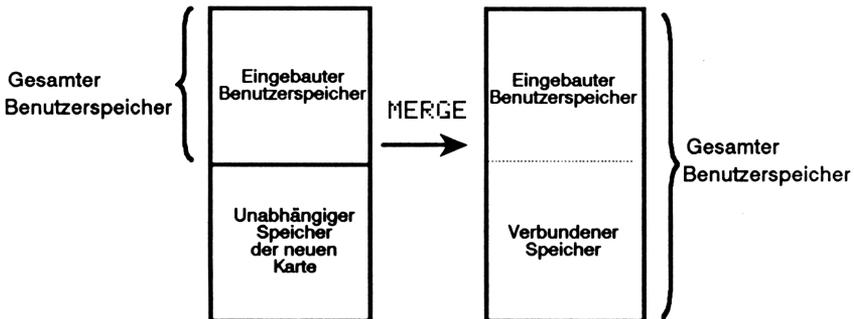


Verwendung von RAM-Karten zur Erweiterung des Benutzerspeichers (Eingebundener Speicher)

Bevor Sie eine installierte RAM-Karte dazu verwenden können, den Benutzerspeicher zu erweitern, muß der Befehl **MERGE** ausgeführt werden. Dadurch wird der Speicher in den internen Speicher *eingebunden*.

Bevor der Befehl **MERGE** ausgeführt wird, muß der Schreibschutzschalter auf der RAM-Karte in der Stellung "Lesen/Schreiben" stehen. (Einzelheiten zur Einstellung des Schreibschutzschalters finden Sie auf Seite 696.)

MERGE verwendet eine Portnummer als Argument. Zum Beispiel wird durch Betätigung der Tasten 1 **MEMORY** **NXT** **NXT** **MERG** der Speicher auf der Karte in Port 1 in den internen Speicher eingebunden.



Wenn Sie eine RAM-Karte einbinden, auf der Sicherungsobjekte gespeichert sind, werden diese Objekte zu einem speziellen Port mit der Bezeichnung Port 0 verschoben. (Port 0 ist auf Seite 702 beschrieben.)



Vorsicht

Auf keinen Fall dürfen Sie eine RAM-Karte ausbauen, die einen Teil des Benutzerspeichers enthält, weil sonst die Daten im Benutzerspeicher verloren gehen können.

Bevor Sie die RAM-Karte herausnehmen, müssen Sie den eingebundenen Speicher *freigeben*. (Einzelheiten hierzu finden Sie im Kapitel "Freigeben des eingebundenen Speichers" auf Seite 704.)

Wenn Sie versehentlich eine Karte mit einem eingebundenen Speicher aus dem Taschenrechner herausnehmen und die Meldung `Replace RAM, Press ON`, sehen, können Sie einen größeren Datenverlust vermeiden, indem Sie *den Taschenrechner eingeschaltet lassen*, die Karte wieder in dasselbe Port einsetzen und dann **ON** drücken.

Verwendung von RAM-Karten für die Sicherung von Daten (Unabhängiger Speicher)

Der HP 48 verwendet einen speziellen Objekttyp, das *Sicherungsobjekt*, um gesicherte Daten zu speichern. Ein Sicherungsobjekt enthält ein anderes Objekt, seinen Namen, und seine Prüfsumme. Einfach ausgedrückt: Ein Sicherungsobjekt enthält eine Variable oder ein Verzeichnis und die zugehörige Prüfsumme.

Eine RAM-Karte mit einem unabhängigen Speicher, welche die Sicherungsobjekte enthält, kann aus dem HP 48 herausgenommen und entweder für spätere Anwendungen aufbewahrt oder in einem anderen HP 48 installiert werden.

Sichern von Objekten in einen unabhängigen Speicher

Sicherungsobjekte können vorkommen

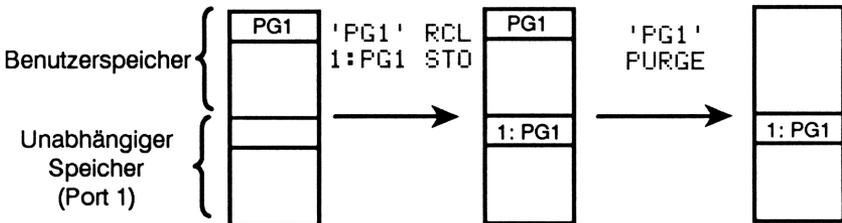
- In einem unabhängigen Speicher (Port 1 und/oder Port 2).
- In einem Bereich des Anwenderspeichers mit der Bezeichnung Port 0 (Siehe Sie dazu Seite 702).

Um ein Sicherungsobjekt zu erstellen, verwenden Sie den Befehl `STO`; dazu sind zwei Argumente erforderlich — das zu sichernde Objekt in Ebene 2 und ein *Sicherungskennzeichen* in Ebene 1. Ein Sicherungskennzeichen hat folgende Form:

`:port#:name`

wobei *port#* die Portnummer (0, 1 oder 2) ist und *name* der Name, unter dem die Sicherungskopie gespeichert wird.

Beispiel: Sichern eines Programms. Um ein Programm mit dem Namen *PG1* im unabhängigen Speicher in Port 1 abzulegen, müssen Sie das Programm durch `'PG1' RCL` auf den Stack rufen und das Objekt anschließend durch `:1:PG1 STO` als Sicherungsobjekt in Port 1 speichern.



Das Sicherungsobjekt im vorigen Beispiel hat denselben Namen wie das Originalobjekt; die beiden Namen dürfen jedoch auch verschieden sein.

Beachten Sie, daß ein Verzeichnis und seine Unterverzeichnisse in einem einzigen Sicherungsobjekt gesichert werden können.

Beispiel: Sichern eines Verzeichnisses und seiner Unterverzeichnisse. Nehmen Sie an, Ihr Verzeichnis *HOME* hat ein Unterverzeichnis mit dem Namen *CHEM*, das selbst weitere Unterverzeichnisse enthält. Sie können das gesamte Verzeichnis *CHEM* mit allen Unterverzeichnissen als Sicherungsobjekt mit dem Namen *BCHEM* speichern; rufen Sie das Verzeichnis durch `'CHEM' RCL` auf den Stack und speichern Sie es anschließend als Sicherungsobjekt, indem Sie `:1:BCHEM STO` eingeben.

Zugreifen auf Sicherungsobjekte

Sie können den Inhalt von Sicherungsobjekten aufrufen, verarbeiten und löschen. Sie können außerdem eine Liste aller Sicherungsobjekte in einem bestimmten Port erstellen.

Aufrufen von Sicherungsobjekten. Sie können das Menü LIBRARY dazu verwenden, den Inhalt von Sicherungsobjekten aufzurufen. Wenn Sie  LIBRARY und anschließend PORT0, PORT1 oder PORT2 drücken, wird ein Menü der Sicherungsobjekte und Bibliotheken in diesem Port angezeigt. Um den Inhalt eines Sicherungsobjektes auf den Stack zu rufen, müssen Sie einfach  und anschließend die Menütaste für das gewünschte Sicherungsobjekt drücken.

Der Befehl RCL kann auch dazu verwendet werden, den Inhalt eines Sicherungsobjektes auf den Stack zu rufen. Wenn Sie zum Beispiel die Sequenz : 1: BPG1 RCL eingeben, wird das in 1: BPG1 gespeicherte Objekt aufgerufen.

Auswerten von Sicherungsobjekten. Wenn Sie das Menü LIBRARY dazu verwenden wollen, den Inhalt eines Sicherungsobjektes auszuwerten, müssen Sie  LIBRARY und anschließend PORT0, PORT1 oder PORT2 drücken. Drücken Sie danach einfach die Menütaste für das gewünschte Sicherungsobjekt.

Auch wenn Sie EVAL drücken und das Argument ein Sicherungsname ist, wird der Inhalt des Sicherungsobjektes ausgewertet. Bei Ausführung der Sequenz : 1: BPG1 EVAL wird zum Beispiel das im Sicherungsobjekt 1: BPG1 gespeicherte Programm ausgewertet (EVAL akzeptiert auch eine Liste von Sicherungsobjekten als Argument, und verarbeitet mehrere Objekte gemeinsam.)

Löschen von Sicherungsobjekten. Wenn Sie ein Sicherungsobjekt löschen wollen, müssen Sie den Sicherungsnamen als Argument für den Befehl PURGE verwenden, zum Beispiel : 1: BPG1 PURGE. (PURGE akzeptiert auch eine Liste von Sicherungsobjekten als Argument und löscht mehrer Objekte gleichzeitig.)

Verwendung von Stellvertreterzeichen bei RCL, EVAL und PURGE.

Das Zeichen & kann als Stellvertreterzeichen für die Portnummer in den Argumenten für RCL, EVAL und PURGE verwendet werden. (& ist **[ENTER]** als links-umgeschaltete alphabetische Taste.) Wenn der HP 48 bei Ausführung dieser Befehle auf das Stellvertreterzeichen trifft, sucht er in den Ports 2, 1, 0 und anschließend im Hauptspeicher nach dem angegebenen Sicherungsobjekt (der Befehl wird mit dem Objekt ausgeführt, dessen Namen zuerst auftritt). Bei Ausführung der Sequenz `:&:BPG1 PURGE` sucht der HP 48 zum Beispiel in Port 2, 1, 0 und anschließend im Hauptspeicher nach dem ersten Erscheinen von *BPG1* und löscht das Objekt anschließend.

Auflisten von Sicherungsobjekten. Der Befehl PVARs (**[←] [MEMORY] [NXT] PVARs**) kann dazu verwendet werden, eine Liste von Objekten in einem bestimmten Port anzuzeigen. Er verwendet eine der Portnummern 0, 1 oder 2 als Argument und legt die Speicherart in dem Port in Ebene 1 ab ("ROM", "SYSRAM" oder eine Zahl, die die freien Speicherplätze im unabhängigen RAM-Speicher darstellt). In Ebene 2 legt er eine Liste der Sicherungsobjekte und der Identifizierungsnummern der Bibliotheken ab (beide markiert durch die Portnummer).

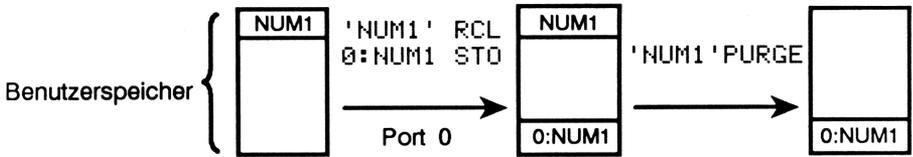
Sie können auch das Menü LIBRARY dazu verwenden, ein Menü von Sicherungsobjekten und Bibliotheken in einem bestimmten Port anzuzeigen. Dazu brauchen Sie nur **[←] [LIBRARY]** und danach **PORT0**, **PORT1** oder **PORT2** zu drücken, um das gewünschte Menü darzustellen.

Sichern von Daten in den Benutzerspeicher (Port 0)

Der HP 48 erlaubt es, Sicherungsobjekte im Benutzerspeicher anzulegen. Der Bereich, der im Benutzerspeicher für Sicherungsobjekte und Bibliotheken verwendet wird, heißt "Port 0". Es gibt verschiedene Gründe, Daten in den Benutzerspeicher zu sichern:

- Sie wollen Daten "geheimhalten"; das bedeutet, daß Sie bestimmte Daten im Benutzerspeicher benötigen, deren Variable aber in *keinem* Inhaltsverzeichnis erscheinen sollen.
- Sie wollen eine RAM-Karte freigeben, die bisher als eingebundener Speicher verwendet worden ist und sie stattdessen als unabhängigen Speicher einsetzen. (Einzelheiten hierzu finden Sie im Abschnitt "Freigeben des eingebundenen Speichers" auf Seite 704).

Sie können ein Sicherungsobjekt im Benutzerspeicher auf dieselbe Weise anlegen wie andere Sicherungsobjekte; Sie müssen nur Port 0 im Sicherungskennzeichen angeben.



Die Größe von Port 0 ist dynamisch: es paßt sich an seinen Inhalt an.

Sichern des gesamten Speichers

Der Befehl ARCHIVE (MEMORY **NXT** **NXT** ARCHI) erzeugt im unabhängigen Speicher ein Sicherungsobjekt mit dem Namen :port#:name; dieses Sicherungsobjekt enthält eine Kopie folgender Objekte:

- Vollständiges Inhaltsverzeichnis *HOME*.
- Benutzerdefinierte Tastenzuweisungen
- Terminkatalog.

Er verwendet einen Namen als Argument, der mit einer Portnummer (0, 1 oder 2) markiert ist. Zum Beispiel wird durch :2:JUN12 ARCHIVE das Sicherungsobjekt :2:JUN12 angelegt.

Mit dem Befehl RESTORE (MEMORY **NXT** **NXT** RESTO) lassen sich die Daten wiederherstellen, die mit dem Befehl ARCHIVE gesichert worden sind. Er verwendet ebenfalls einen Namen als Argument, der durch eine Portnummer markiert ist; dabei muß das entsprechende Objekt ein Inhaltsverzeichnis sein. Mit der Sequenz :2:JUN12 RESTORE stellen Sie zum Beispiel das Verzeichnis *HOME* wieder her, das im obigen Beispiel gesichert wurde.



Vorsicht

Die Ausführung von RESTORE *überschreibt* den gesamten Inhalt des Benutzerspeichers mit dem Inhalt des Sicherungsobjektes.

Wenn Sie Ihre Flag-Einstellungen beim Sichern des gesamten Speichers ebenfalls "retten" wollen, müssen Sie sie mit RCLF auf den Stack zurückrufen und in einer Variablen speichern, bevor Sie den Befehl ARCHIVE geben. Nach Wiederherstellung des Speichers mit RESTORE können Sie die Flag-Einstellungen wieder aktivieren, indem Sie den Inhalt der Variablen auf den Stack rufen und den Befehl STOF geben (Flags speichern).

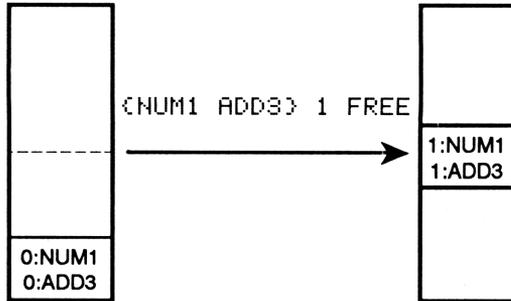
Freigeben des eingebundenen Speichers

Das Freigeben eines eingebundenen Speichers wandelt ihn in einen unabhängigen Speicher um. Ein eingebundener Speicher muß freigegeben werden, wenn

- Sie die RAM-Karte aus ihrem Port herausnehmen wollen.
- Sie die RAM-Karte als unabhängigen Speicher und nicht als Teil des Benutzerspeichers verwenden wollen.

Der Befehl FREE ( MEMORY  NXT  FREE) gibt den eingebundenen Speicher in einem angegebenen Port frei. Er verwendet zwei Argumente — eine Liste in Ebene 2 und die Portnummer in Ebene 1.

Die Liste kann leer sein; dann wird der eingebundene Speicher einfach freigegeben. Sie kann aber auch einen oder mehrere Namen enthalten oder auch Identifizierungskennzeichen von Bibliotheken. In diesem Fall verschiebt FREE die angegebenen Sicherungsobjekte und Bibliotheken aus Port 0 in die neu freigegebene Karte. Zum Beispiel gibt die Ausführung der Sequenz `{ NUM1 ADD3 } 1 FREE` den eingebundenen Speicher in Port 1 frei und macht ihn zu einem unabhängigen Speicher. Gleichzeitig werden die Sicherungsobjekte *NUM1* und *ADD3* von Port 0 nach Port 1 verschoben.



Um einen eingebundenen Speicher freizugeben, müssen Sie zunächst mit MEM die Größe des verfügbaren Speichers bestimmen (Drücken Sie **MEMORY** **MEM**). Ist die Größe des verfügbaren Speichers größer oder gleich dem Speicherbereich auf der Karte, den Sie freigeben wollen, können Sie den Befehl FREE geben.

Wenn MEM einen Wert liefert, der kleiner ist als der Speicherbereich auf der Karte, erhalten Sie bei Ausführung von FREE eine Fehlermeldung, wenn Sie nicht bestimmte Vorbereitungen treffen. Ihre gespeicherten Daten passen dann nämlich nicht in den Speicherbereich, der nach der Freigabe des eingebundenen Speichers noch zur Verfügung steht. Sie können folgendes tun, um eine Fehlermeldung zu vermeiden:

- Löschen Sie Variablen aus dem Benutzerspeicher, die Sie nicht mehr benötigen.
- Sichern Sie Daten auf einer anderen RAM-Karte, die in dem anderen Port installiert ist, und löschen Sie anschließend die Originalvariablen.
- Sichern Sie Daten in Port 0 (interner Speicher) und schieben Sie diese Daten anschließend, indem Sie sie in Ebene 2 als Argument für den Befehl FREE angeben, in den freigegebenen Speicher. Dies können Sie erreichen, wenn Sie die folgenden Schritte ausführen:
 1. Bestimmen Sie die Datenmenge, die in den Speicher zu verschieben ist, den Sie freigeben wollen. Wollen Sie zum Beispiel eine Karte mit 128 KByte freigeben, und im Benutzerspeicher stehen 126 KByte zur Verfügung, müssen Sie Variablen, die insgesamt mindestens 2 KByte belegen, verschieben.

2. Sichern Sie die Variablen in Port 0. Wenn Sie zum Beispiel *CALC1* in Port 0 sichern wollen, müssen Sie ihren Inhalt auf den Stack rufen und `:0:CALC1 STO` ausführen.
3. Löschen Sie die Variable aus dem Anwenderspeicher (zum Beispiel: `'CALC' PURGE`).
4. Sichern und löschen Sie, falls erforderlich, weitere Variablen und Inhaltsverzeichnisse.
5. Wenn Sie genügend Daten gesichert haben, können Sie den Befehl `FREE` geben. Das Argument in Ebene 2 muß eine Liste mit den Namen der Variablen und Inhaltsverzeichnisse sein, die Sie in Port 0 gesichert haben.

Verwenden von ROM-Steckkarten und Bibliotheken

Eine *Bibliothek* ist ein Objekt, das benannte Objekte enthält, mit denen der interne Befehlssatz erweitert werden kann. Den Inhalt einer Bibliothek können Sie weder im Anzeigefeld darstellen noch verändern. Bibliotheken können auf ROM-Steckkarten gespeichert sein oder ins RAM kopiert werden. Bibliotheken lassen sich allerdings nicht mit dem HP 48 anlegen.

Bibliotheken besitzen:

- Ein *Identifizierungskennzeichen für eine Bibliothek* mit dem Aufbau `:port#:library#`. Die *Bibliotheksnummer library#* ist eine wichtige Zahl, die mit der Bibliothek verknüpft ist. Das Identifizierungskennzeichen für die Bibliothek wird als Argument für Befehle verwendet, die die Objekte einer Bibliothek verarbeiten.
- Einen *Bibliotheksnamen*, der aus einer Zeichenfolge besteht. Der Name einer Bibliothek erscheint im Menü `LIBRARY`, wenn die Bibliothek an ein Inhaltsverzeichnis im aktuellen Pfad angebunden wird.

Anbinden einer Bibliothek an ein Inhaltsverzeichnis

Wenn Sie mit einer Bibliothek arbeiten wollen, muß sie an ein Inhaltsverzeichnis im Benutzerspeicher *angebunden* sein. Das Anbinden kann automatisch bei der Installation der Karte erfolgen; es ist auch möglich, daß Sie es selbst ausführen müssen. Lesen Sie bitte die Dokumentation Ihrer ROM-Steckkarte (oder Ihrer RAM-Bibliothek), in der die Vorgehensweise zum Anbinden der Bibliothek beschrieben ist.

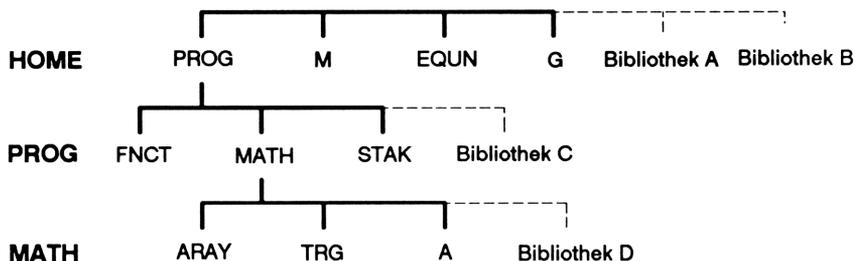
Wenn die Bibliothek nicht automatisch angebunden wird, müssen Sie den Befehl ATTACH ( MEMORY  ATTACH) verwenden. ATTACH benötigt eine Bibliotheksnummer als Argument.

Es können beliebig viele Bibliotheken an das Inhaltsverzeichnis HOME angebunden werden. Es kann jedoch immer nur eine Bibliothek an ein bestimmtes Unterverzeichnis angebunden werden.

Zugreifen auf Bibliotheksoperationen (Das Menü LIBRARY)

Das Menü LIBRARY. Nach Betätigen von  LIBRARY wird das Menü LIBRARY angezeigt, das die Namen der Bibliotheken im aktuellen Verzeichnispfad enthält. Wenn Sie ein Menü der Operationen in einer Bibliothek anzeigen wollen, müssen Sie die entsprechende Taste drücken. Haben sie zum Beispiel eine Gleichungsbibliothek für HP SOLVE in Ihrem Taschenrechner installiert, wird nach Betätigen von  LIBRARY  EQLTIE ein Menü aller Operationen in dieser Bibliothek angezeigt.

Zugreifen auf Bibliotheken, die an Unterverzeichnisse angebunden sind. Für den Zugriff auf Bibliotheken, die an Unterverzeichnisse angebunden sind, gelten dieselben Regeln wie für den Zugriff auf Variablen in diesen Verzeichnissen. Nehmen Sie zum Beispiel an, in Ihrem HP 48 besteht folgende Verzeichnisstruktur mit angehängten Bibliotheken:



Wenn *HOME* das aktuelle Verzeichnis ist, wird nach Betätigen von **LIBRARY** das Menü seiner angebundenen Bibliotheken angezeigt — . Wenn *PROG* das aktuelle Verzeichnis ist, wird nach Drücken von **LIBRARY** ein Menü mit der an *PROG* angebundenen Bibliothek , sowie mit den anderen Bibliotheken im aktuellen Pfad und dargestellt.

Wie auf Variablen kann auch auf Operationen in einer Bibliothek immer dann zugegriffen werden, wenn die Bibliothek an das aktuelle Verzeichnis oder an ein Verzeichnis im aktuellen Pfad angebunden ist. Zum Beispiel können Sie auf die Operationen der Bibliotheken A und B von jedem beliebigen Verzeichnis aus zugreifen, da diese beiden Bibliotheken an *HOME* angebunden sind. Sie können auf die Operationen in der Bibliothek C zugreifen, wenn *PROG* oder *MATH* das aktuelle Verzeichnis sind. Sie können allerdings nicht auf die Operationen in Bibliothek D zugreifen, wenn *PROG* das aktuelle Verzeichnis ist.

Weitere Befehle für den Zugriff auf Bibliotheken

Alle folgenden Befehle mit Ausnahme von LIBS und ATTACH verwenden das Identifizierungskennzeichen (:port#:library#) als Argument.

Bibliotheksbefehle

Tasten	Programmierbarer Befehl	Beschreibung
STO	STO	Speichert eine Bibliothek in einem unabhängigen Speicher in einem angegebenen Port.
RCL	RCL	Ruft eine Bibliothek auf den Stack.
PURGE	PURGE	Löscht eine angegebene RAM-Bibliothek.
MEMORY (Seite 2):		
PVARS	PVARS	Zeigt eine Liste an, die die Identifizierungskennzeichen für Sicherungsobjekte und für Bibliotheken in einem angegebenen Port enthält.
LIBS	LIBS	Zeigt eine Liste an, die die Namen, Bibliotheksnummern und Portnummern aller Bibliotheken enthält, die an das aktuelle Verzeichnis angebunden sind.
ATTAC	ATTACH	Bindet eine angegebene Bibliothek an das aktuelle Verzeichnis an.
DETAC	DETACH	Koppelt eine angegebene Bibliothek vom aktuellen Verzeichnis ab.

Teil 6

Anhänge und Verzeichnisse

Unterstützung, Batterien und Wartung

Unterstützung bei der Arbeit mit dem Taschenrechner

Alle Fragen, die bei der Arbeit mit Ihrem Taschenrechner auftauchen, beantworten Ihnen die Mitarbeiter unserer Abteilung für Kundenbetreuung. Nach unserer Erfahrung haben viele Kunden gleiche oder ähnliche Fragen zu unseren Produkten; aus diesem Grund haben wir den folgenden Abschnitt “Antworten auf häufige Fragen” in das Handbuch aufgenommen. Wenn Sie hier keine Antwort finden, können Sie Ihre Fragen auch schriftlich oder telefonisch an uns richten. Unsere Anschrift und die Rufnummer finden Sie hinten in diesem Handbuch auf der Innenseite des Einbandes.

Antworten auf häufige Fragen

F: *Der Taschenrechner läßt sich nicht einschalten, wenn ich drücke. Was ist nicht in Ordnung?*

A: Die Ursache kann ein einfaches Problem sein, das Sie sehr schnell selbst lösen können; es ist aber auch möglich, daß der Taschenrechner zur Reparatur eingeschickt werden muß. Einzelheiten finden Sie im Abschnitt “Überprüfung der Taschenrechnerfunktion” auf Seite 722.

F: *Ich bin nicht sicher, ob der Taschenrechner nicht richtig funktioniert oder ob ich einen Fehler bei der Bedienung mache. Wie kann ich überprüfen, ob der Taschenrechner fehlerfrei arbeitet?*

A: Lesen Sie bitte den Abschnitt “Überprüfung der Taschenrechnerfunktion” auf Seite 722 in diesem Anhang.

F: Der Indikator (•) bleibt sichtbar, auch wenn der Taschenrechner ausgeschaltet ist. Ist irgendetwas nicht in Ordnung?

A: Das deutet auf eine zu niedrige Batteriespannung im Taschenrechner oder auf einer RAM-Karte hin; es kann auch auf einen überfälligen Termin hinweisen. Schalten Sie den Taschenrechner aus und wieder ein, um die Ursache für das ständig sichtbare Symbol (•) festzustellen. Im Anzeigefeld erscheint eine Meldung, die die Störung bezeichnet. Weitere Einzelheiten finden Sie im Kapitel "Batteriewechsel" auf Seite 718 in diesem Anhang oder im Kapitel "Einstellung von Terminen" auf Seite 479 in Kapitel 24.

F: Wie kann ich feststellen, wieviel Speicherplatz im Rechner mir noch zur Verfügung steht?

A: Drücken Sie  **MEMORY** . Die Anzahl der Bytes im verfügbaren Speicher erscheint dann in der rechten unteren Ecke des Anzeigefeldes. Bei einem leeren Speicher muß hier ungefähr 30000 (Bytes des internen RAM) angezeigt werden.

F: Wie kann ich den gesamten Speicher des Taschenrechners löschen?

A: Führen Sie die folgenden Schritte aus:

1. Drücken Sie **ON** und halten Sie die Taste fest.
2. Drücken Sie gleichzeitig die beiden äußeren Tasten in der oberen Reihe (die Menütasten neben den Zeichen A und F) und lassen Sie diese gleichzeitig wieder los.
3. Lassen Sie **ON** los.

Das akustische Signal des Taschenrechners ertönt, und die Frage Try To Recover Memory? ("Versuchen, Speicherinhalt wieder herzustellen?") erscheint in der Anzeige. Drücken Sie **NO**, um den Anwenderspeicher zu löschen; es erscheint dann die Meldung Memory Clear ("Speicher gelöscht") in der Anzeige.



Hinweis

Auf diese Weise löschen Sie nicht den Schreib-Lese-Speicher auf der Einsteckkarte, außer wenn Sie diesen RAM-Speicher in den Hauptspeicher des Taschenrechners eingebunden haben.

F: *Wie ändere ich die Anzahl der Dezimalstellen, die der HP 48 anzeigt?*

A: Führen Sie die folgenden Schritte aus:

1. Rufen Sie Seite 1 im Menü MODES auf: Drücken Sie  **MODES**.
2. Geben Sie über die Zifferntasten die benötigte Anzahl der Dezimalstellen ein (0 — 11).
3. Drücken Sie die Menütaste für das gewünschte Anzeigeformat (**FIX**, **SCI**, or **ENG**).

Lesen Sie den Abschnitt “Anzeigemodi” in Kapitel 2 (Seite 61).

F: *Meine Zahlen enthalten Kommata zur Darstellung der Dezimalstellen. Wie kann ich wieder auf die Punktdarstellung umschalten?*

A: Führen Sie die folgenden Schritte aus:

1. Rufen Sie Seite 4 im Menü MODES auf (Drücken Sie  **MODES** **NXT** **NXT** **NXT**).
2. Drücken Sie die Menütaste zur Umschaltung des Dezimalzeichens **FM, ■**. (Das ■ muß nach Betätigen der Menütaste verschwinden.)

F: *Was bedeutet ein “E” in einer Zahl (zum Beispiel: 2,51E-13) ?*

A: Das “E” kennzeichnet den Exponenten von 10 (zum Beispiel: $2,51 \times 10^{-13}$). Einzelheiten finden Sie im Abschnitt “Anzeigemodi” auf Seite 61 in Kapitel 2.

F: *Warum erhalte ich ‘SIN(π)’ anstelle einer Zahl, wenn ich im Gradmodus den Sinus von π bestimme?*

A: Der Taschenrechner arbeitet im Modus zur Darstellung von symbolischen Ergebnissen; ‘SIN(π)’ ist das symbolische Ergebnis. Drücken Sie  **NUM**, um ‘SIN(π)’ in seinen Zahlenwert ,0548... mit maximal 11 Dezimalstellen umzuwandeln. Sie können auch **SYM** auf Seite 1 des Menüs MODES drücken, um den Modus zur Darstellung von Zahlenwertergebnissen aufzurufen und damit die symbolische Darstellung auszuschalten.

F: *Was bedeutet “Objekt” ?*

A: “Objekt” ist der allgemeine Ausdruck für alle Datenelemente, mit denen der HP 48 arbeitet. Zahlen, mathematische Ausdrücke, Felder, Programme, auch Grafiken, all dies sind verschiedene Arten von Objekten. Im Kapitel 4 “Objekte” sind die Objekttypen beschrieben, die der Taschenrechner verwendet.

F: Was bedeuten drei Punkte (...) an beiden Enden einer Zeile in der Anzeige ?

A: Die drei Punkte (bezeichnet als *ellipsis*) bedeuten, daß das angezeigte Objekt zu lang für die Anzeige in einer Zeile ist. Die nichtangezeigten Anteile des Objektes können Sie durch Betätigen der Cursortasten ◀ oder ▶ darstellen.

F: Das akustische Signal des Taschenrechners ertönt, und in der Anzeige erscheint Bad Argument Type ("Falscher Argumenttyp"). Was ist nicht richtig?

A: Die Objekte im Stack entsprechen nicht dem Typ, den Sie für die Ausführung des gewünschten Befehls benötigen. Zum Beispiel führt die Ausführung von →UNIT (auf Seite 2 des Menüs PRG OBJ) mit einer Zahl in den Stackebenen 1 und 2 zu diesem Fehler.

F: Das akustische Signal des Taschenrechners ertönt, und in der Anzeige erscheint Too Few Arguments ("Zu wenige Argumente"). Was ist nicht richtig?

A: Es sind weniger Argumente im Stack vorhanden, als Sie für die Ausführung des gewünschten Befehls benötigen. Zum Beispiel führt die Ausführung von [+] mit nur einem Argument oder einer Zahl auf dem Stack zu diesem Fehler.

F: Das akustische Signal des Taschenrechners ertönt und in der Anzeige erscheint eine andere Meldung als die zuvor beschriebenen Meldungen. Wie kann ich feststellen, wo der Fehler liegt?

A: Lesen Sie den Abschnitt "Meldungen" in Anhang B.

F: Ich kann einige Variable, die ich zu einem früheren Zeitpunkt verwendet habe, nicht wiederfinden. Wohin können sie verschwunden sein?

A: Es ist möglich, daß Sie die Variablen in einem anderen Datenverzeichnis verwendet haben. Wenn Sie sich nicht mehr an das verwendete Verzeichnis erinnern können, müssen Sie alle Dateiverzeichnisse in Ihrem Taschenrechner durchsehen.

F: *Manchmal scheint der HP 48 bei der Durchführung von Berechnungen eine Pause von einigen Sekunden zu machen. Deutet das auf einen Fehler hin?*

A: Nein. Von Zeit zu Zeit organisiert der Taschenrechner seinen Speicher neu und beseitigt dabei vorübergehend benötigte Objekte, die bei normalen Rechenoperationen entstehen. Dadurch entsteht freier Speicherplatz für die gerade ausgeführten Operationen.

F: *Während des normalen Betriebs werden einige Zeilen vom Drucker sehr schnell gedruckt; danach wird er langsamer. Warum?*

A: Der Taschenrechner überträgt eine bestimmte Datenmenge sehr schnell zum Drucker und verringert dann seine Übertragungsgeschwindigkeit, damit der Drucker Schritt halten kann.

F: *Wie kann ich die Druckgeschwindigkeit meines Infrarotdruckers HP 82240B erhöhen?*

A: Verwenden Sie einen AGC-Adapter für Ihren Drucker HP 82240B, so daß der Drucker schneller arbeiten kann. Passen Sie außerdem an Ihrem Taschenrechner die Verzögerungszeit zwischen zwei Zeilen an die Druckgeschwindigkeit an (Einzelheiten hierzu finden Sie im Abschnitt "Einstellen der Verzögerungszeit" auf Seite 657).

Umgebungsbedingungen

Taschenrechner. Um die zuverlässige und fehlerfreie Funktion Ihres Taschenrechners zu gewährleisten, müssen Sie dafür sorgen, daß das Gerät nicht feucht wird und daß die folgenden Höchstwerte für die Umgebungstemperatur und die relative Luftfeuchtigkeit nicht überschritten werden:

- Betriebstemperatur: 0° bis 45°C.
- Lagertemperatur: -20° bis 65°C.
- Luftfeuchtigkeit bei Betrieb und Lagerung: Maximal 90% relative Feuchtigkeit bei 40°C.

Einsteckkarten. Für die Einsteckkarten von Hewlett Packard gelten folgende Grenzwerte für die Umgebungsbedingungen :

- Betriebstemperatur: 0° bis 45°C.
- Lagertemperatur: –20° bis 60°C.
- Lagertemperatur für die Aufbewahrung von RAM-Karten: 0° bis 60°C.
- Luftfeuchtigkeit bei Betrieb und Lagerung: Maximal 90% relative Feuchtigkeit bei 40°C.

Was beim Austausch der Batterien zu beachten ist

Bei einer zu niedrigen Batteriespannung bleibt der Indikator (••) sichtbar; auch wenn der Taschenrechner ausgeschaltet ist. Wird der Taschenrechner bei einer zu niedrigen Batteriespannung eingeschaltet, erscheint für ungefähr 3 Sekunden die Anzeige `Warning: LowBat()` (“Achtung: zu niedrige Batteriespannung”). `LowBat(P1)` bezieht sich auf Port 1, `LowBat(P2)` gilt für Port 2 und `LowBat(S)` weist auf die (System-) Batterien des Taschenrechners hin.

Wechseln Sie die RAM-Kartenbatterie oder die Batterien des Taschenrechners möglichst bald, wenn der Indikator (••) und die Warnungsmeldung für zu niedrige Batteriespannung erscheinen. Wenn Sie trotz der (••) Meldung mit Ihrem Taschenrechner weiterarbeiten, wird die Anzeige immer schwächer, und es ist möglich, daß Daten im Rechner und auf der RAM-Karte verloren gehen.

Unter den üblichen Betriebsbedingungen hält eine RAM-Kartenbatterie zwischen 1 und 3 Jahren. Denken Sie daran, das Datum auf der Karte zu notieren, an dem Sie die Batterie eingesetzt haben. Ist die RAM-Karte nicht im Rechner installiert, wenn die Batterie ausgetauscht werden muß, sollten Sie sich durch einen Termin 1 Jahr nach diesem Datum daran erinnern lassen, eine neue Batterie einzusetzen. RAM-Karten werden ohne Batterie geliefert.

Batterien wechseln

Batterietypen

Batterien des Taschenrechners. Batterien der Größe AAA eines beliebigen Fabrikates. *Achten Sie darauf, daß alle drei Batterien von demselben Hersteller und vom gleichen Typ sind.*

Von wiederaufladbaren Batterien wird abgeraten, da diese eine geringere Kapazität haben.

Batterien für Einsteck-RAM-Karten. Knopfzelle 2016 (3 Volt)

Batterien des Taschenrechners wechseln

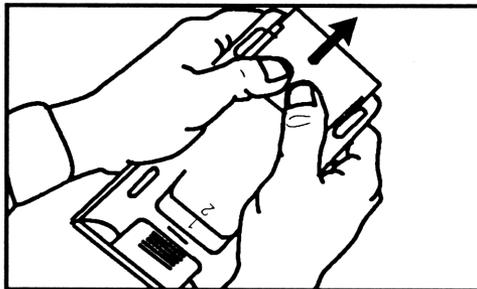
Diese Hinweise gelten für das Wechseln der *Taschenrechner*-Batterien. Hinweise zum Austausch der RAM-Kartenbatterien finden Sie ab Seite 720.



Vorsicht

Achten Sie immer darauf, daß der Taschenrechner ausgeschaltet ist, wenn Sie die Batterien aus dem Taschenrechner herausnehmen. *Drücken Sie die Taste **[ON]** erst dann, wenn Sie die neuen Batterien eingesetzt haben.* Wenn Sie die Taste **[ON]** betätigen, obwohl sich keine Batterien im Taschenrechner befinden, können alle Daten im Speicher des Rechners verloren gehen.

1. Schalten Sie den Taschenrechner aus. Der Inhalt des Speichers im Taschenrechner und auf der RAM-Karte kann verloren gehen, wenn Sie die Batterien im eingeschalteten Zustand aus dem Gerät herausnehmen.
2. Legen Sie drei neue Batterien desselben Herstellers und vom gleichen Typ bereit. Wischen Sie die beiden Pole jeder einzelnen Batterie mit einem sauberen, trockenen Tuch ab.
3. Nehmen Sie die Abdeckung des Batteriefaches heraus: Drücken Sie, wie auf der folgenden Abbildung dargestellt, die Abdeckung leicht nach innen und schieben Sie sie aus dem Rechnergehäuse heraus. Achten Sie darauf, daß Sie dabei nicht auf die **[ON]**-Taste des Gerätes drücken.



4. Drehen Sie den Taschenrechner um, und entfernen Sie die Batterien durch leichtes Schütteln aus ihrem Fach. Wenn Sie die Batterien herausgenommen haben, sollten Sie sie innerhalb von 2 Minuten durch neue ersetzen, um einen Verlust von Speicherdaten zu vermeiden.



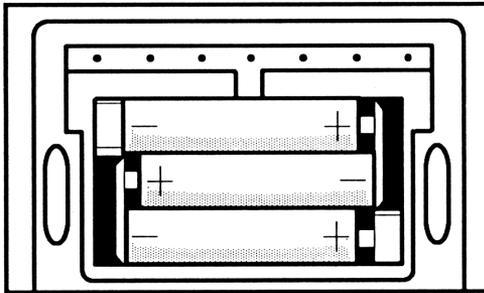
Warnung

Verbrauchte Batterien dürfen nicht zerstört oder offenem Feuer ausgesetzt werden. Die Batterien können platzen oder explodieren und dabei gefährliche chemische Substanzen freisetzen.

Beachten Sie bei der Entsorgung verbrauchter Batterien die Hinweise des Herstellers.

5. *Vermeiden Sie es, die Batterieanschlüsse zu berühren* Die Batterien lassen sich leichter einbauen, wenn Sie die (flache) Seite mit dem negativen Pol zuerst einsetzen und die mittlere Batterie als letzte installieren.

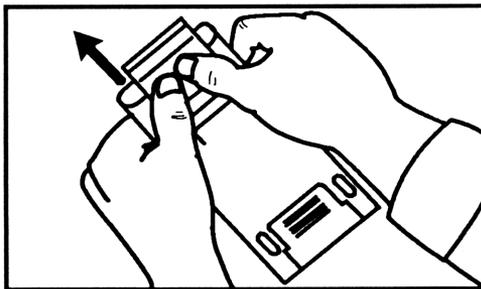
Beachten Sie beim Einbau der Batterien hinsichtlich der Pole das Schema am Boden des Batteriefaches. Es entspricht der folgenden Abbildung:



6. Schieben Sie die Abdeckung des Batteriefaches mit den Laschen in die Führung im Rechnergehäuse.
7. Drücken Sie **[ON]**, um den Taschenrechner einzuschalten.

Batterie der RAM-Karte wechseln

1. Drehen Sie den Taschenrechner um und nehmen Sie die Kunststoffabdeckung über der Einsteckkarte ab (an dem Ende des Taschenrechners, an dem sich das Anzeigefeld befindet).



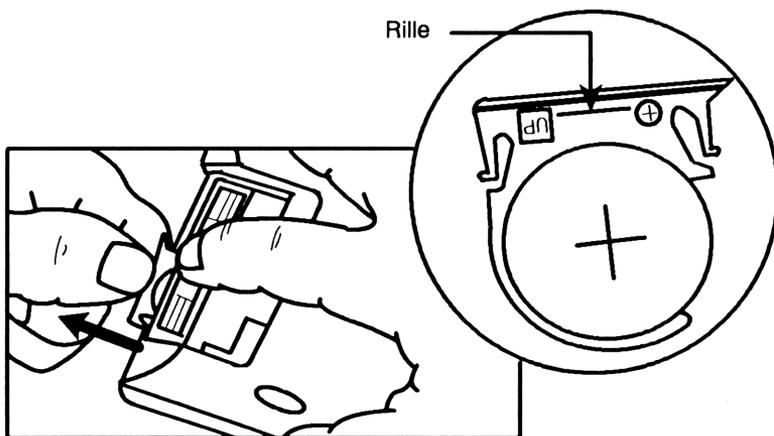
2. Schalten Sie den Taschenrechner ein; die RAM-Karte muß sich dabei in Port 1 oder 2 befinden.



Vorsicht

Da RAM-Karten über die Batterien des Taschenrechners versorgt werden, wenn das Gerät eingeschaltet ist (Schalterstellung **[ON]**), dürfen Sie die Batterie einer solchen Karte nur austauschen, wenn die Karte im Rechner installiert und der Taschenrechner eingeschaltet ist. Der Speicherinhalt des RAM kann verloren gehen, wenn Sie eine RAM-Kartenbatterie bei ausgeschaltetem Taschenrechner ersetzen.

3. Legen Sie Ihren Zeigefinger in die Vertiefung an der offenen Seite der RAM-Karte — dadurch verhindern Sie, daß die Karte beim Herausnehmen der Batteriehalterung ebenfalls aus dem Rechner herausgezogen wird. Fassen Sie dann mit der freien Hand den schwarzen Batteriehalter und ziehen Sie ihn aus der Karte heraus. Vorne an dem Halter ist eine Rille für Ihren Daumnagel vorgesehen.



4. Nehmen Sie die alte Batterie aus der Kunststoff-Batteriehalterung heraus.



Warnung

Verbrauchte Batterien dürfen nicht zerstört oder offenem Feuer ausgesetzt werden. Die Batterien können platzen oder explodieren und dabei gefährliche chemische Substanzen freisetzen.

Beachten Sie bei der Entsorgung verbrauchter Batterien die Hinweise des Herstellers.

5. Setzen Sie eine neue 3-Volt-Knopfzelle 2016 in den Kunststoff-Batteriehalter ein, und schieben Sie den Batteriehalter (mit Batterie) wieder in die RAM-Karte. *Achten Sie darauf, daß die mit “+” markierte Seite der Batterie zur vorderen Seite der Karte hin zeigt (siehe Abbildung oben).*
6. Notieren Sie das Datum auf der Karte, an dem Sie die Batterie eingesetzt haben. Stellen Sie zusätzlich einen Termin ein, der Sie ein Jahr nach diesem Datum daran erinnert, daß die Batterie gewechselt werden muß. Dies ist wichtig für den Fall, daß die Karte nicht in Ihrem Rechner installiert sein sollte.
7. Setzen Sie die Abdeckung über dem Anschluß für die Einsteckkarte wieder ein.

Überprüfung der Taschenrechnerfunktion

Führen Sie die im folgenden beschriebenen Bedienungsschritte aus, wenn Sie überprüfen wollen, ob Ihr Taschenrechner fehlerfrei funktioniert. Testen Sie Ihren Taschenrechner nach jedem einzelnen Schritt, um festzustellen, ob er wieder einwandfrei arbeitet. Wenn Ihr Taschenrechner gewartet werden muß, finden Sie die entsprechenden Hinweise auf Seite 732.

Der Taschenrechner läßt sich nicht einschalten oder reagiert nicht auf Tastendrucke.

1. Falls Sie die Batterien gewechselt haben, kontrollieren Sie, ob die neuen Batterien richtig eingebaut sind.
2. Wenn das Anzeigefeld leer ist, müssen Sie **[ON]** drücken und festhalten. Drücken Sie nun mehrfach **[+]**, bis Zeichen auf der Anzeige sichtbar werden. Lassen Sie anschließend **[ON]** los. Sind jetzt keine Zeichen mehr im Anzeigefeld zu sehen, muß der Rechner zur Reparatur eingeschickt werden.
3. Wenn ein unterbrochenes Programm nicht auf die Betätigung der Taste **[ATTN]** reagiert, versuchen Sie es noch einmal.
4. Wenn die Tastatur "gesperrt" ist, können Sie eine Systemunterbrechung mit folgenden Schritten herbeiführen:
 - a. Drücken Sie **[ON]** und halten Sie die Taste gedrückt.
 - b. Drücken Sie die dritte Taste von links in der oberen Reihe (die Menütaste neben dem C), und lassen Sie sie wieder los.
 - c. Lassen Sie **[ON]** wieder los.

Es muß eine leere Stackanzeige erscheinen.

5. Erscheinen im Anzeigefeld unsinnige Zeichen, ist der Speicher auf folgende Weise zurückzusetzen:
 - a. Drücken Sie **[ON]** und halten Sie die Taste gedrückt.
 - b. Drücken Sie die beiden äußeren Tasten in der oberen Reihe (die Menütasten neben dem A und dem F) und halten Sie beide Tasten gedrückt.
 - c. Lassen Sie alle drei Tasten los.

Es ertönt das akustische Signal des Taschenrechners, und es erscheint die Anzeige Try To Recover Memory? ("Speicherinhalt wieder herzustellen versuchen?") oben im Anzeigefeld. Drücken Sie **[YES]**, um soviel wie möglich vom Speicherinhalt wiederherzustellen .

Läßt sich mit den beschriebenen Schritten eine störungsfreie Funktion des Taschenrechners nicht wiederherstellen, muß das Gerät repariert werden.

Der Taschenrechner reagiert auf die Bedienung der Tasten; Sie vermuten jedoch, daß er fehlerhaft funktioniert.

1. Lassen Sie den Selbsttest laufen, der im nächsten Abschnitt beschrieben ist. Tritt beim Selbsttest ein Fehler auf, muß der Taschenrechner repariert werden.
2. Läuft der Selbsttest fehlerfrei ab, kann die Ursache der Störung ein Bedienungsfehler sein. Lesen Sie entsprechenden Abschnitte des Handbuches noch einmal, und beachten sie auch die Hinweise im Abschnitt "Antworten auf allgemeine Fragen" (Seite 712).
3. Wenden Sie sich an unsere Abteilung zur Kundenbetreuung für Taschenrechner. Anschrift und Rufnummer finden Sie hinten in diesem Handbuch auf der Innenseite des Einbandes.

Selbsttest

Wenn sich das Anzeigefeld einschalten läßt, der Taschenrechner aber nicht fehlerfrei zu funktionieren scheint, sollten Sie den folgenden Selbsttest zur Diagnose laufen lassen:

1. Drücken Sie **[ON]** und halten Sie die Taste fest.
2. Drücken Sie die zweite Taste von rechts in der oberen Reihe (die Menütaste neben E), und lassen Sie diese Taste wieder los.
3. Lassen Sie **[ON]** los.

Bei diesem Selbsttest zur Diagnose werden die internen RAM- und ROM-Speicher überprüft und außerdem verschiedene Muster erzeugt und im Anzeigefeld dargestellt. Die Prüfung wird bis zu ihrem Abbruch kontinuierlich wiederholt.

4. Um einen Selbsttest abzubrechen, können Sie eine Systemunterbrechung mit den folgenden Schritten herbeiführen:
 - a. Drücken Sie **[ON]** und halten Sie die Taste gedrückt.
 - b. Drücken Sie die dritte Taste von links in der oberen Reihe (die Menütaste neben dem C), und lassen Sie die Taste wieder los.
 - c. Lassen Sie **[ON]** los.

Es muß eine leere Stackanzeige erscheinen.

Der Selbsttest zur Diagnose muß fehlerfrei abgelaufen sein, bevor irgendeiner der anderen Tests in den folgenden Abschnitten ausgeführt wird.

Weist der Selbsttest auf einen Fehler der internen RAM- oder ROM-Speicher hin (wenn also IROM OK und IRAM OK nicht angezeigt werden), muß der Taschenrechner zur Reparatur eingeschickt werden.

Testen der Tastatur

Mit diesem Test soll festgestellt werden, ob alle Tasten des Taschenrechners fehlerfrei funktionieren.

Um die interaktive Tastaturprüfung auszuführen, sind folgende Bedienungsschritte erforderlich:

1. Drücken Sie **[ON]** und halten Sie die Taste fest.
2. Drücken Sie die dritte Taste in der oberen Reihe (die Menütaste neben dem D), und lassen Sie die Taste wieder los.
3. Lassen Sie **[ON]** los.
4. Drücken Sie die zweite Taste in der oberen Reihe (die Menütaste neben E) und lassen Sie die Taste wieder los. In der linken oberen Ecke des Anzeigefeldes erscheint `ƘED1`.
5. Drücken Sie nun nacheinander die 49 Tasten auf der Tastatur. Beginnen Sie in der oberen linken Ecke und gehen Sie dabei von links nach rechts. Wenn Sie die Tasten in der richtigen Reihenfolge drücken und sie ohne Fehler funktionieren, ertönt nach jedem Tastendruck ein hochfrequentes akustisches Signal. Wenn Sie die 49. Taste gedrückt haben, muß in der Anzeige `ƘED1 OK` erscheinen.

Wenn Sie eine der Tasten außer der Reihe betätigen, erscheint neben `ƘED1` eine fünfstellige Hexadezimalzahl. Brechen Sie die Prüfung der Tastatur ab (führen Sie dazu die oben beschriebenen Schritte 1 bis 3 aus) und beginnen Sie den Test von vorn.

Wenn eine Taste nicht richtig funktioniert, wird mit dem nächsten Tastendruck die erwartete und die empfangene hexadezimale Adresse angezeigt. Wenn Sie die Tasten in der richtigen Reihenfolge gedrückt haben und trotzdem diese Meldung erhalten, muß der Rechner repariert werden. Eine Beschreibung des Fehlers und die bei der Tastaturprüfung angezeigte Fehlermeldung sollten Sie den Unterlagen beifügen, wenn Sie das Gerät zur Reparatur einschicken.

6. Um die Prüfung der Tastatur zu beenden, können Sie mit den folgenden Bedienungsschritten eine Systemunterbrechung herbeiführen.
 - a. Drücken Sie **[ON]** und halten Sie die Taste gedrückt.
 - b. Drücken Sie die dritte Taste von links in der oberen Reihe (die Menütaste neben dem C), und lassen Sie die Taste wieder los.
 - c. Lassen Sie **[ON]** los.

Es muß eine leere Stackanzeige erscheinen.

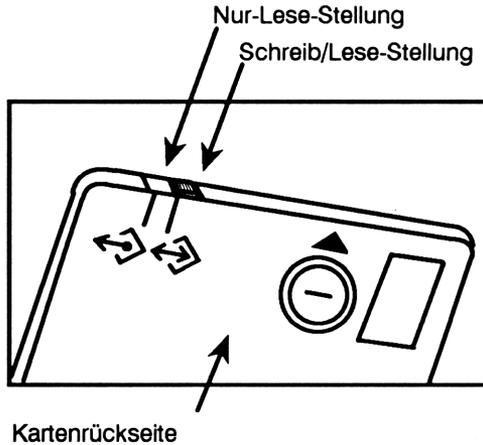
Testen des Port-RAM

Durch den Test des Port-RAM können Sie die feststellen, ob die Ports und die installierten Einsteckkarten fehlerfrei arbeiten. (Die Speicherinhalte der RAM-Einsteckkarten bleiben dabei erhalten.)

Folgende Schritte sind zum Testen des Port-RAM auszuführen:

1. Kontrollieren Sie, ob eine RAM-Einsteckkarte richtig in Port 1 und/oder Port 2 installiert ist.

2. Prüfen Sie, ob auf jeder Karte der Schalter in der Stellung "Schreiben/Lesen" steht:



3. Schalten Sie den Taschenrechner ein.
4. Drücken Sie **[ON]** und halten Sie die Taste fest.
5. Drücken Sie die vierte Taste von links in der oberen Reihe (die Menütaste neben dem D), und lassen Sie die Taste wieder los.
6. Lassen Sie **[ON]** los.

Eine vertikale Linie erscheint an beiden Seiten und in der Mitte des Anzeigefeldes.

7. Drücken Sie **[▲]** und lassen Sie die Taste wieder los.

In der oberen linken Ecke des Anzeigefeldes erscheint **RAM1** und/oder **RAM2**; die Speichergröße der entsprechenden RAM-Einsteckkarte (32K oder 128K) wird in der rechten oberen Ecke angezeigt. Wenn der RAM-Test fehlerfrei abgelaufen ist, wird das durch die Meldung **OK** rechts neben **RAM1** und/oder **RAM2** bestätigt. Für jeden Port ohne RAM-Einsteckkarte erscheint eine Fehlermeldung (zum Beispiel **RAM1 00002**). Sie erscheint auch dann, wenn der Schreib-Lese-Schalter der Karte in der Stellung "Schreibschutz" steht. Diese Meldung können Sie ignorieren.

Wenn die Meldung **OK** nicht erscheint, obwohl die RAM-Karte auf "Schreiben/Lesen" eingestellt ist, muß die Karte mit dem anderen Port verbunden und der Test neu gestartet werden. Wird die Meldung **OK** nach Ablauf der Prüfung immer noch nicht angezeigt, muß die RAM-Karte gegen eine neue Karte ausgetauscht werden.

8. Um zur normalen Funktion des Taschenrechners zurückzukehren, können Sie mit den folgenden Bedienungsschritten eine Systemunterbrechung herbeiführen:

- a. Drücken Sie **[ON]** und halten Sie die Taste fest.
- b. Drücken Sie die dritte Taste von links in der oberen Reihe (die Menütaste neben dem C), und lassen Sie die Taste wieder los.
- c. Lassen Sie **[ON]** los.

Es muß eine leere Stackanzeige erscheinen.

Schleifentest des IR-Anschlusses

Mit diesem Test werden die Funktion der Sende- und Empfangssensoren für die IR-Übertragung und die zugehörigen elektronischen Schaltkreise geprüft.

Dazu sind folgende Bedienungsschritte erforderlich:

1. Drücken Sie **[ON]** und halten Sie die Taste fest.
2. Drücken Sie die vierte Taste von links in der oberen Reihe (die Menütaste neben dem D), und lassen Sie die Taste wieder los.

3. Lassen Sie **[ON]** los. Eine vertikale Linie erscheint an beiden Seiten und in der Mitte des Anzeigefeldes.
4. Achten Sie darauf, daß die Kunststoffabdeckung der Einsteckkarte sich an ihrem Platz befindet und daß sie die beiden durchsichtigen Lampen am oberen Ende des Taschenrechners verdeckt.
5. Drücken Sie die Taste **[EVAL]**.

In der oberen linken Ecke des Bildschirms wird **IRLB** angezeigt. Die Meldung **OK** erscheint rechts neben **IRLB**, wenn dieser Test des Taschenrechners ohne Fehler abgelaufen ist.

Wenn die Meldung **OK** nicht erscheint, muß der Taschenrechner zur Reparatur eingeschickt werden.

6. Um zur normalen Funktion des Taschenrechners zurückzukehren, können Sie mit den folgenden Bedienungsschritten eine Systemunterbrechung herbeiführen.
 - a. Drücken Sie **[ON]** und halten Sie die Taste fest.
 - b. Drücken Sie die dritte Taste von links in der oberen Reihe (die Menütaste neben dem C), und lassen Sie die Taste wieder los.
 - c. Lassen Sie **[ON]** los.

Es muß eine leere Stackanzeige erscheinen.

Schleifentest des seriellen Anschlusses

Mit diesem Test wird die Funktion der Sende- und Empfangsschaltungen des seriellen Anschlusses am oberen Ende des Taschenrechners überprüft.

Dazu sind folgende Bedienungsschritte erforderlich:

1. Drücken Sie **[ON]** und halten Sie die Taste fest.
2. Drücken sie die vierte Taste von links in der oberen Reihe (die Menütaste neben dem D), und lassen Sie die Taste wieder los.
3. Lassen Sie **[ON]** wieder los. Eine vertikale Linie erscheint an beiden Seiten und in der Mitte des Anzeigefeldes.

4. Verbinden Sie für den Test die beiden mittleren Stifte (Kurzschluß der Stifte 2 und 3) des 4-poligen, seriellen Steckverbinders am oberen Ende des Taschenrechners. Achten Sie darauf, daß die Anschlußstifte nicht verbogen oder abgebrochen werden.
5. Drücken Sie die Taste **PRG**.

In der linken oberen Ecke des Anzeigefeldes erscheint die Meldung U_LB.

Wenn diese Prüfung des Taschenrechners ohne Fehler abläuft, erscheint die Meldung OK rechts neben U_LB.

Erscheint die Meldung OK nicht, muß der Taschenrechner zur Reparatur eingeschickt werden.



Hinweis

Wenn Sie versehentlich die Anschlußstifte 1 und 2 oder 3 und 4 des seriellen Steckverbinders kurzschließen, meldet der Schleifentest U_LB 00001 oder U_LB 00002 (Fehlermeldung bei der Prüfung); der Taschenrechner wird dadurch jedoch nicht beschädigt.

6. Um zur normalen Funktion des Taschenrechners zurückzukehren, können Sie mit den folgenden Bedienungsschritten eine Systemunterbrechung herbeiführen:
 - a. Drücken Sie **ON** und halten Sie die Taste fest.
 - b. Drücken Sie die dritte Taste von links in der oberen Reihe (die Menütaste neben dem C), und lassen Sie die Taste los.
 - c. Lassen Sie **ON** wieder los.

Es muß eine leere Stackanzeige erscheinen. Vergessen Sie nicht, den Kurzschluß am seriellen Anschluß wieder zu entfernen.

Begrenzte Ein-Jahres-Garantie

Was wird von der Garantie abgedeckt. Hewlett-Packard übernimmt bei dem *Taschenrechner und seinem Zubehör* die Garantie für die Ausführung von Reparaturen und die erforderlichen Ersatzteile während eines Zeitraums von einem Jahr vom Tag der Anschaffung durch den Kunden an. Von dieser Garantie ausgenommen sind die Batterien oder Schäden, die durch die Batterien verursacht werden. *Wenn Sie Ihr Gerät verkaufen oder verschenken, gehen die Garantieansprüche im Garantiezeitraum von einem Jahr automatisch auf den neuen Besitzer über.*

Während der Garantiezeit wird ein defektes Gerät von uns repariert oder kostenlos gegen ein anderes Gerät ausgetauscht. Voraussetzung ist, daß Sie das Produkt auf Ihre Kosten an Ihr Hewlett-Packard-Reparaturzentrum senden. (Wir behalten uns vor, ein defektes Gerät gegen ein neueres Modell mit gleichen oder verbesserten Funktionen auszutauschen.)

Was wird nicht von der Garantie übernommen. *Batterien und Schäden, die durch die Batterien verursacht werden, sind nicht durch die Garantie von Hewlett-Packard gedeckt. Bitte beachten Sie die Garantiebedingungen des Batterieherstellers.*

Die Garantie verfällt, wenn das Produkt versehentlich oder durch Einsatz unter Bedingungen, für die es nicht vorgesehen ist, beschädigt wurde oder wenn ein Schaden bei Wartung oder Veränderung durch einen nicht von Hewlett-Packard autorisierten Reparaturbetrieb entstanden ist.

Andere Garantieleistungen sind ausgeschlossen. Sie haben Rechtsanspruch ausschließlich auf Reparatur oder Austausch eines Produktes.

HEWLETT-PACKARD IST IN KEINEM FALL FÜR FOLGESCHÄDEN VERANTWORTLICH.

Beim Verkauf der Produkte sind die technischen Daten zum Zeitpunkt der Herstellung maßgebend. Hewlett-Packard ist nicht dazu verpflichtet, bereits verkaufte Produkte zu modifizieren oder auf den neuen technischen Stand zu bringen.

Wenn der Taschenrechner repariert oder gewartet werden muß



Hinweis

Wenn Sie in Ihrem Taschenrechner wichtige Daten gespeichert haben, sollten Sie diese Daten auf einer RAM-Einsteckkarte, in einem anderen HP 48 oder in einem Computer speichern, bevor Sie das Gerät zur Reparatur an Hewlett-Packard einschicken.

Hewlett-Packard hat Reparaturzentren in vielen Ländern eingerichtet. In diesen Reparaturzentren wird der Taschenrechner repariert oder durch das gleiche Modell mit gleichen oder verbesserten Funktionen ausgetauscht; unabhängig davon, ob die Garantiezeit abgelaufen ist oder nicht. Nach Ablauf der Garantiezeit werden Wartungs- oder Reparaturkosten berechnet. Die Taschenrechner werden in der Regel innerhalb von 5 Tagen repariert oder gewartet und an den Kunden zurückgesandt.

- **In Europa:** Wenden Sie sich an Ihr Verkaufsbüro von Hewlett-Packard, an Ihren Händler oder an die europäische Zentrale (Anschrift siehe unten), um die Adresse des nächsten Wartungszentrums zu erfahren. *Setzen Sie sich zuerst mit einer Geschäftsstelle von Hewlett-Packard in Verbindung, bevor Sie den Taschenrechner zur Reparatur einsenden.*

Hewlett-Packard S.A.
150, Route du Nant-d'Avril
P.O. Box CH 1217 Meyrin 2
Geneva, Switzerland
Telephone: 022 780.81.11

- **Deutschland:** Wenden Sie sich direkt an das Reparaturzentrum in Frankfurt:

Reparaturzentrum Frankfurt
Berner Straße 117
6000 Frankfurt 56
Tel. (069) 5 00 06-0

- **In anderen Ländern:** Wenden Sie sich an Ihr Verkaufsbüro von Hewlett-Packard, an Ihren Händler oder schreiben Sie an das Corvallis Service Center, um die Anschrift anderer Wartungszentren zu erfahren. Wenn eine Reparatur oder Wartung in Ihrer Nähe nicht möglich ist, können Sie das Gerät an das Corvallis Service Center senden.

Alle Formalitäten beim Versand und für die Wiedereinfuhr müssen vom Kunden geregelt werden. Der Kunde hat auch die Zollgebühren zu bezahlen.

Versandvorschriften. Senden Sie Ihren Taschenrechner an das nächste autorisierte Reparaturzentrum oder an die nächste Geschäftsstelle, wenn das Gerät repariert werden muß.

- Geben Sie die Rücksendeadresse an und legen Sie eine Fehlerbeschreibung bei.
- Legen Sie Ihre Garantiekarte mit Angabe des Kaufdatums bei, wenn die Garantiezeit noch nicht abgelaufen ist.
- Versenden Sie Ihren Taschenrechner auf *Ihre* Kosten und in einer geeigneten Schutzverpackung, um Beschädigungen beim Transport zu vermeiden. Beschädigungen beim Transport werden nicht von den Garantieleistungen gedeckt; deshalb ist eine Transportversicherung zu empfehlen.

Garantie auf Wartungsleistungen. Bei Wartungsleistungen wird eine Garantie auf Material und Ausführung für einen Zeitraum von 90 Tagen vom Tag der Wartung an übernommen.

Störsicherheit

USA. Der HP 48 erzeugt und arbeitet mit hochfrequenten Signalen; aus diesem Grund ist es möglich, daß Rundfunk- und Fernsehempfang gestört werden. Der Taschenrechner hält die Grenzwerte für Rechner der Klasse B entsprechend den Angaben in Abschnitt J von Teil 15 der FCC-Vorschriften ein, die einen ausreichenden Schutz gegen derartige Störstrahlungen in Wohnräumen bieten. Sollte der Rundfunk- oder Fernsehempfang wider Erwarten gestört werden (was durch Ein- und Ausschalten des HP 48 oder durch Herausnehmen der Batterien festgestellt werden kann), können Sie folgendes versuchen:

- Richten Sie die Empfangsantenne anders aus.
- Wählen Sie einen anderen Standort für Ihren Taschenrechner in bezug auf den Empfänger.

Bundesrepublik Deutschland. Der HP 48 erfüllt die Anforderungen VFG 1046/84, VDE 0871B und entsprechender Normen und Richtlinien zur Vermeidung von Störstrahlung. Wenn Sie den Taschenrechner in Geräteaufbauten einsetzen, die nicht den Vorschriften von Hewlett-Packard entsprechen, muß der Systemaufbau den Anforderungen in Paragraph 2 des Bundesgesetzblattes 1046/84, vom 14. Dezember 1984 entsprechen.

Meldungen

In diesem Anhang sind ausgewählte Meldungen des HP 48 zusammengestellt.

In den folgenden Tabellen sind die Meldungen zuerst in alphabetisch und anschließend nach ihrer Meldungsnummer geordnet.

Alphabetisch zusammengestellte Meldungen

Meldung	Bedeutung	# (hex)
Acknowledged	Alarm bestätigt.	619
Autoscaling	Der Taschenrechner skaliert automatisch die x- und/oder y-Achse.	610
Awaiting Server Cmd.	Server-Modus aktiv.	C0C
Bad Argument Type	Der Typ eines oder mehrerer Stack-Parameter ist mit der durchgeführten Operation nicht vereinbar.	202
Bad Argument Value	Parameterwert außerhalb des Gültigkeitsbereiches.	203
Bad Guess(es)	Schätzwert für HP-SOLVE oder ROOT liegt außerhalb des Gleichungsbereiches.	A01

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Bad Packet Block check	Die berechnete Prüfsumme des Paketes stimmt nicht mit der Prüfsumme im Paket überein.	C01
Can't Edit Null Char.	Versuch, eine Zeichenkette mit dem Zeichen char# 0 zu editieren.	102
Circular Reference	Versuch, einen Variablennamen in sich selbst zu speichern.	129
Connecting	Zeigt die Prüfung der Infrarot- oder der seriellen Verbindung an.	C0A
Constant?	HP-SOLVE oder ROOT haben an jedem Punkt der aktuellen Gleichung denselben Wert geliefert.	A02
Copied to stack	→STK hat die ausgewählte Gleichung in den Stack kopiert.	623
Current equation:	Kennzeichnet die aktuelle Gleichung.	608
Deleting Column	Der MatrixWriter löscht gerade eine Spalte.	504
Deleting Row	Der MatrixWriter löscht gerade eine Zeile.	503
Directory Not Allowed	Der Name einer existierenden Verzeichnisvariablen wird als Argument verwendet.	12A
Directory Recursion	Versuch, ein Verzeichnis in sich selbst zu speichern.	002
Empty catalog	Keine Daten im aktuellen Katalog (Gleichung, Statistik, Termin)	60D

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Enter alarm, press SET	Aufforderung zur Alarmeingabe.	61A
Enter eqn, press NEW	Neue Gleichung in EQ speichern.	60A
Enter value (zoom out if >1), press ENTER	Aufforderung zur Ausführung der Zoom-Funktion.	622
Extremum	Das von HP-SOLVE oder von ROOT zurückübertragene Ergebnis ist ein Extremwert und keine Nullstelle der Funktion.	A06
HALT Not Allowed	Ein Programm mit der Anweisung HALT wurde ausgeführt, während der MatrixWriter, DRAW oder die HP-SOLVE aktiv waren.	126
I/O setup menu	Kennzeichnet das Menü zur Einstellung der E/A-Parameter.	61C
Implicit () off	Implizite Klammern ausgeschaltet.	207
Implicit () on	Implizite Klammern eingeschaltet.	208
Incomplete Subexpression	 ,  , oder ENTER gedrückt, bevor alle Parameter einer Funktion eingegeben waren .	206
Inconsistent Units	Versuch einer Umwandlung mit nicht passenden Einheiten.	B02
Infinite Result	Mathematischer Fehler: Berechnung mit einer unendlichen Größe als Ergebnis, zum Beispiel die Division 1/0.	305

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Inserting Column	Der MatrixWriter fügt gerade eine Spalte ein.	504
Inserting Row	Der MatrixWriter fügt gerade eine Zeile ein.	503
Insufficient Memory	Nicht genügend freier Speicherplatz zur Ausführung einer Funktion.	001
Insufficient Σ Data	Ein Statistikbefehl wurde ausgeführt, ΣDAT enthielt jedoch nicht genügend Datenpunkte zur Berechnung.	603
Interrupted	HP-SOLVE oder ROOT wurden durch [ATTN] unterbrochen.	A03
Invalid Array Element	[ENTER] hat Daten des falschen Typs geliefert, sie passen nicht in die aktuelle Matrix.	502
Invalid Card Data	Der HP 48 erkennt Daten auf der Steckkarte nicht.	008
Invalid Date	Ein Argument ist keine reelle Zahl im richtigen Format oder liegt außerhalb des Bereiches.	D01
Invalid Definition	Falscher Aufbau einer Gleichung für DEFINE.	12C
Invalid Dimension	Ein Feldargument hatte falsche Dimensionen.	501

Alphabetisch zusammengestellte (Meldungen)

Meldung	Bedeutung	# (hex)
Invalid EQ	Versuch einer Bedienung aus dem Menü GRAPHICS FCN, wobei EQ keinen algebraischen Ausdruck enthielt oder Versuch, DRAW in der Darstellungsart CONIC auszuführen, wobei EQ keinen algebraischen Ausdruck enthielt.	607
Invalid IOPAR	<i>IOPAR</i> enthält keine Liste, oder ein oder mehrere Parameter in der Liste fehlen oder sind ungültig.	C12
Invalid Name	Empfang eines unzulässigen Dateinamens oder das Steuergerät hat zum Senden eines unzulässigen Dateinamens aufgefordert.	C17
Invalid PPAR	<i>PPAR</i> enthält keine Liste, oder ein oder mehrere Parameter in der Liste fehlen oder sind ungültig.	12E
Invalid PRTPAR	<i>PRTPAR</i> enthält keine Liste, oder ein oder mehrere Parameter in der Liste fehlen oder sind ungültig.	C13
Invalid PTYPE	Plottyp für die aktuelle Gleichung ist ungültig.	620
Invalid Repeat	Wiederholungsintervall für die Terminerinnerung liegt außerhalb des zulässigen Bereiches .	D03
Invalid Server Cmd.	Empfang eines ungültigen Befehls im Server-Modus.	C08
Invalid Syntax	Der HP 48 kann [ENTER] or STR→ nicht ausführen wegen ungültiger Syntax eines Objektes.	106

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Invalid Time	Der Zeitparameter ist keine reelle Zahl im richtigen Format oder liegt außerhalb des zulässigen Bereiches.	D02
Invalid Unit	Versuch einer Operation mit einer ungültigen oder nichtdefinierten Benutzereinheit.	B01
Invalid User Function	Art oder Aufbau des Objektes, das als benutzerdefinierte Funktion ausgeführt wurde, war falsch.	103
Invalid Σ Data	Versuch, einen Statistikbefehl auszuführen, wobei ein ungültiges Objekt in ΣDAT gespeichert war.	601
Invalid Σ Data LN(Neg)	Versuch, eine nichtlineare Kurve zu ermitteln, wobei ΣDAT ein negatives Element enthielt.	605
Invalid Σ Data LN(0)	Versuch, eine nichtlineare Kurve zu ermitteln, wobei die Matrix ΣDAT ein 0-Element enthielt.	606
Invalid ΣPAR	ΣPAR enthält keine Liste oder ein oder mehrere Parameter in der Liste fehlen oder sind ungültig.	604
LAST CMD Disabled	[LAST CMD] wurde gedrückt, obwohl diese Funktion gesperrt war.	125
LAST STACK Disabled	[LAST STACK] wurde gedrückt, obwohl diese Funktion gesperrt war.	124
LASTARG Disabled	LASTARG wurde ausgeführt, obwohl diese Funktion gesperrt war.	205

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Low Battery	Die Batteriespannung des Systems ist zu gering, um fehlerfrei zu drucken oder eine Eingabe/Ausgabe auszuführen.	C14
Memory Clear	Der Speicher wurde gelöscht.	005
Name Conflict	Versuch, (wobei) auszuführen, um einer Integrationsvariablen oder einem Summierungsindex einen Wert zuzuweisen.	13C
Name the equation, press ENTER	Der Gleichung einen Namen zuweisen und in EQ speichern.	60B
Name the stat data, press ENTER	Den Statistikdaten einen Namen zuweisen und in ΣDAT speichern.	621
Negative Underflow	Mathematischer Fehler: Die Berechnung lieferte ein negatives Ergebnis ungleich Null, das größer als -MINR ist.	302
No Current Equation	SOLVR, DRAW, oder RCEQ wurde auszuführen versucht, ohne daß EQ existierte.	104
No current equation	Statusmeldung von PLOT oder HP-SOLVE.	609
No Room in Port.	Nicht genügend freier Speicher im angegebenen RAM-Port.	00B
No Room to Save Stack	Nicht genug freier Speicher, um die Kopie des Stacks zu sichern. LAST STACK wird automatisch gesperrt.	101

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
No Room to Show Stack	Objekte im Stack werden wegen nicht ausreichendem Speicherplatz nur durch ihren Typ dargestellt.	131
No stat data to plot	Keine Daten in Σ DAT gespeichert.	60F
Non-Empty Directory	Versuch, ein nicht leeres Verzeichnis zu löschen.	12B
Non-Real Result	Die Ausführung von HP-SOLVE, von ROOT, DRAW oder <i>f</i> lieferte ein anderes Ergebnis als eine reelle Zahl oder eine Einheit.	12F
Nonexistent Alarm	Ein Termin, der durch einer Terminbefehl angegeben wurde, ist in der Terminliste nicht enthalten	D04
Nonexistent Σ DAT	Ein Statistikbefehl wurde ausgeführt, obwohl Σ DAT nicht existierte.	602
Object Discarded	Der Sender hat ein EOF-(Z)-Paket mit einem "D" im Datenfeld übertragen.	C0F
Object In Use	Versuch, PURGE oder STO auf ein Sicherungsobjekt anzuwenden, als der zugehörige gespeicherte Datenblock bearbeitet wurde.	009
Object Not in Port	Versuch, auf ein nichtvorhandenes Sicherungsobjekt oder eine Bibliothek zuzugreifen.	00C

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
(OFF SCREEN)	Funktionswert, Nullstelle, Extremwert oder Schnittpunkt waren innerhalb des aktuellen Anzeigefeldes nicht sichtbar.	61F
Out of Memory	Eines oder mehrere Objekte müssen gelöscht werden, um den Betrieb des Taschenrechners fortsetzen zu können.	135
Overflow	Mathematischer Fehler: Die Berechnung lieferte ein Ergebnis, dessen absoluter Wert größer als MAXR ist.	303
Packet #	Gibt die Paketnummer während des Sendens oder beim Empfang an.	C10
Parity Error	Das Paritätsbit der empfangenen Bytes entspricht nicht der aktuellen Paritätseinstellung.	C05
Port Closed	Mögliche Gerätestörung bei Infrarot- oder serieller Schnittstelle. Selbsttest ausführen.	C09
Port Not Available	Es wurde ein Portbefehl auf ein leeres Port angewendet oder ein Befehl ausgeführt, der auf das ROM statt auf das RAM zugreift. Versuch, einen Server-Befehl auszuführen, der selbst den E/A-Anschluß verwendet.	00A

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Positive Underflow	Mathematischer Fehler: Die Berechnung lieferte ein positives Ergebnis ungleich Null, das kleiner als MINR ist.	301
Power Lost	Der Taschenrechner wurde nach einem Spannungsausfall wieder eingeschaltet. Es ist möglich, daß der Speicherinhalt verfälscht ist.	006
Processing Command	Meldet, daß ein Befehlspaket des Leitrechners verarbeitet wird.	C11
Protocol Error	Empfang eines Paketes, dessen Länge kürzer war als ein Nullpaket.	C07
	Der Parameter für die maximale Paketlänge des anderen Rechners ist ungültig.	
Receive Buffer Overrun	Kermit: Es wurden mehr als 255 wiederholte Bytes gesendet, bevor der HP 48 ein neues Paket empfangen hat.	C04
	SRECV: Die ankommenden Daten haben zu einem Überlauf geführt.	
Receive Error	UART-Datenverlust oder Synchronisierfehler.	C03
Receiving	Erkennt den Namen des Objektes während des Empfangs.	C0E
Retry #	Zeigt die Anzahl der Versuche während der Wiederholung des Paketaustausches an.	C0B

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Select a model	Statistikmodell zur Kurvenermittlung wählen.	614
Select plot type	Darstellungsart wählen.	60C
Select repeat interval	Wiederholungsintervall für Terminerinnerung wählen.	61B
Sending	Erkennt den Namen des Objektes während des Sendens.	C0D
Sign Reversal	HP-SOLVE oder ROOT konnte keinen Punkt finden, an dem die verwendete Gleichung zu Null wird; fand aber zwei benachbarte Punkte, zwischen denen die Gleichung das Vorzeichen wechselte.	A05
Timeout	Drucken über den seriellen Anschluß: Empfang von XOFF, Unterbrechung, es wird auf XON gewartet. Kermit: Unterbrochen, es wird auf die Ankunft eines Paketes gewartet.	C02
Too Few Arguments	Der Befehl erforderte mehr Argumente, als im Stack zur Verfügung standen.	201
Transfer Failed	10 aufeinander folgende Versuche, ein geeignetes Paket zu empfangen, sind fehlgeschlagen.	C06

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
Unable to Isolate	ISOL konnte nicht ausgeführt werden, weil der angegebene Name fehlt oder im Argument einer Funktion enthalten ist, die keine Umkehrung besitzt	130
Undefined Local Name	Hat einen lokalen Namen ausgeführt oder aufgerufen, für den eine entsprechende lokale Variable nicht existiert.	003
Undefined Name	Hat einen globalen Namen ausgeführt oder aufgerufen, für den eine entsprechende Variable nicht existiert.	204
Undefined Result	Eine Berechnung wie z.B. 0/0 hat ein mathematisch nicht definiertes Ergebnis geliefert.	304
Undefined XLIB Name	Versuch, einen XLIB-Namen auszuführen, es fehlte jedoch die angegebene Bibliothek.	004
Wrong Argument Count	Anwenderdefinierte Funktion mit einer falschen Anzahl von Klammerausdrücken ausgewertet.	128
x and y-axis zoom.	Kennzeichnet eine Zoom-Option.	627
x axis zoom.	Kennzeichnet eine Zoom-Option.	625
x axis zoom w/AUTO.	Kennzeichnet eine Zoom-Option.	624

Alphabetisch zusammengestellte Meldungen (Fortsetzung)

Meldung	Bedeutung	# (hex)
y axis zoom.	Kennzeichnet eine Zoom-Option.	626
ZERO	Das von HP-SOLVE oder ROOT gelieferte Ergebnis ist eine Nullstelle (der Punkt, an der der Funktionswert der verwendeten Gleichung zu Null wird).	A04
""	Kennzeichnet keine Ausführung, wenn EXECS gedrückt wird.	61E

Numerisch zusammengestellte Meldungen

# (hex)	Meldung
Allgemeine Meldungen	
001	Insufficient Memory
002	Directory Recursion
003	Undefined Local Name
004	Undefined XLIB Name
005	Memory Clear
006	Power Lost
008	Invalid Card Data
009	Object In use
00A	Port Not available
00B	No Room in Port
00C	Object Not in Port
101	No Room to Save Stack
102	Can't Edit Null Char.
103	Invalid User Function
104	No Current Equation

Numerisch sortierte Meldungen (Fortsetzung)

# (hex)	Meldung
Allgemeine Meldungen (Fortsetzung)	
106	Invalid Syntax
124	LAST STACK Disabled
125	LAST CMD Disabled
126	HALT Not Allowed
128	Wrong Argument Count
129	Circular Reference
12A	Directory Not Allowed
12B	Non-Empty Directory
12C	Invalid Definition
12E	Invalid PPAR
12F	Non-Real Result
130	Unable to Isolate
131	No Room to Show Stack
Zu wenig Speicher	
135	Out of Memory
13C	Name Conflict
Fehler im Stack	
201	Too Few Arguments
202	Bad Argument Type
203	Bad Argument Value
204	Undefined Name
205	LASTARG Disabled
Meldungen des EquationWriter	
206	Incomplete Subexpression
207	Implicit () off
208	Implicit () on

Numerisch sortierte Meldungen (Fortsetzung)

# (hex)	Meldung
Gleitkomma-Fehler	
301	Positive Underflow
302	Negative Underflow
303	Overflow
304	Undefined Result
305	Infinite Result
Matrixmeldungen	
501	Invalid Dimension
502	Invalid Array Element
503	Deleting Row
504	Deleting Column
505	Inserting Row
506	Inserting Column
Statistikmeldungen	
601	Invalid Σ Data
602	Nonexistent Σ DAT
603	Insufficient Σ Data
604	Invalid Σ PAR
605	Invalid Σ Data LN(Neg)
606	Invalid Σ Data LN(0)
Meldungen zur grafischen Darstellung, Eingabe/Ausgabe, Zeit und HP-SOLVE	
607	Invalid EQ
608	Current equation:
609	No current equation.
60A	Enter eqn, press NEW
60B	Name the equation, press ENTER
60C	Select plot type
60D	Empty catalog
60F	No Statistics data to plot

Numerisch sortierte Meldungen (Fortsetzung)

# (hex)	Meldung
Meldungen zur grafischen Darstellung, Eingabe/Ausgabe, Zeit und HP-SOLVE (Fortsetzung)	
610	Autoscaling
614	Select a model
619	Acknowledged
61A	Enter alarm, press SET
61B	Select repeat interval
61C	I/O setup menu
61D	Plot type:
61E	""
61F	(OFF SCREEN)
620	Invalid PTYPE
621	Name the stat data, press ENTER
Anwendungsmeldungen	
622	Enter value (zoom out if >1), press ENTER
623	Copied to stack
624	x axis zoom w/AUTO.
625	x axis zoom.
626	y axis zoom.
627	x and y-axis zoom.
A01	Bad Guess(es)
A02	Constant?
A03	Interrupted
A04	Zero
A05	Sign Reversal
A06	Extremum
Einheiten	
B01	Invalid Unit
B02	Inconsistent Units

Numerisch sortierte Meldungen (Fortsetzung)

# (hex)	Meldung
Eingabe/Ausgabe und Drucken	
C01	Bad Packet Block check
C02	Timeout
C03	Receive Error
C04	Receive Buffer Overrun
C05	Parity Error
C06	Transfer Failed
C07	Protocol Error
C08	Invalid Server Cmd
C09	Port Closed
C0A	Connecting
C0B	Retry #
C0C	Awaiting Server Cmd.
C0D	Sending
C0E	Receiving
C0F	Object Discarded
C10	Packet #
C11	Processing Command
C12	Invalid IOPAR
C13	Invalid PRTPAR
C14	I/O: Batt Too Low
C15	Empty Stack
C17	Invalid Name
Zeitmeldungen	
D01	Invalid Date
D02	Invalid Time
D03	Invalid Repeat
D04	Nonexistent Alarm

Zeichencodes des HP 48

Die meisten Zeichen im Zeichensatz des HP 48 können direkt über die Buchstabentasten eingegeben werden.

Wenn Sie zum Beispiel das Zeichen $\$$ darstellen wollen, müssen Sie $\boxed{\text{a}}$ $\boxed{\leftarrow}\boxed{4}$ über die Tastatur eingeben. (Die Eingabe von Buchstaben über die Tastatur ist in Kapitel 2 beschrieben.) Grundsätzlich können Sie jedoch alle Zeichen im Zeichensatz erzeugen, indem Sie den zugehörigen Zeichencode und anschließend den Befehl CHR eingeben. Die Syntax lautet *char#* CHR. Bestimmte Zeichen im Zeichensatz, die sich *nicht* auf der alphabetischen Tastatur befinden, *müssen* Sie also auf jeden Fall auf diese Weise eingeben.

Die Zeichentabellen auf den folgenden Seiten enthalten die Zeichen des HP 48 und die zugehörigen Zeichencodes. (Dieser Zeichensatz basiert, mit Ausnahme der Zeichennummern 128 bis einschließlich 150, auf dem Zeichensatz Latin 1 nach ISO 8859.)



Hinweis

Wenn Sie feststellen, daß ein häufig verwendetes Zeichen sich nicht auf der Primär- oder der Buchstabentastatur befindet (in Kapitel 2 sind alle über die Tastatur erreichbaren Zeichen aufgeführt), können Sie dieses Zeichen für den schnellen Zugriff der Benutzertastatur zuordnen. Einzelheiten hierzu finden Sie im Abschnitt "Umbelegen von Tasten" auf Seite 235.

Zeichencodes (0 — 127)

NUM	ZCH	NUM	ZCH	NUM	ZCH	NUM	ZCH
0	■	32		64	@	96	'
1	■	33	!	65	A	97	a
2	■	34	"	66	B	98	b
3	■	35	#	67	C	99	c
4	■	36	\$	68	D	100	d
5	■	37	%	69	E	101	e
6	■	38	&	70	F	102	f
7	■	39	'	71	G	103	g
8	■	40	<	72	H	104	h
9	■	41	>	73	I	105	i
10	■	42	*	74	J	106	j
11	■	43	+	75	K	107	k
12	■	44	,	76	L	108	l
13	■	45	-	77	M	109	m
14	■	46	.	78	N	110	n
15	■	47	/	79	O	111	o
16	■	48	0	80	P	112	p
17	■	49	1	81	Q	113	q
18	■	50	2	82	R	114	r
19	■	51	3	83	S	115	s
20	■	52	4	84	T	116	t
21	■	53	5	85	U	117	u
22	■	54	6	86	V	118	v
23	■	55	7	87	W	119	w
24	■	56	8	88	X	120	x
25	■	57	9	89	Y	121	y
26	■	58	:	90	Z	122	z
27	■	59	;	91	[123	{
28	■	60	<	92	\	124	
29	■	61	=	93]	125	}
30	■	62	>	94	^	126	~
31	...	63	?	95	_	127	■

Zeichencodes (128 — 255)

NUM	ZCH	NUM	ZCH	NUM	ZCH	NUM	ZCH
128	À	160		192	À	224	à
129	Á	161	í	193	Á	225	á
130	Â	162	ê	194	Â	226	â
131	Ã	163	ë	195	Ã	227	ã
132	Ä	164	ï	196	Ä	228	ä
133	Å	165	ü	197	Å	229	å
134	▶	166	!	198	Æ	230	æ
135	π	167	§	199	Ç	231	ç
136	ò	168	"	200	È	232	è
137	é	169	ó	201	É	233	é
138	ê	170	ä	202	Ê	234	ê
139	ë	171	«	203	Ë	235	ë
140	α	172	¬	204	Ì	236	ì
141	÷	173	–	205	Í	237	í
142	←	174	Ø	206	Î	238	î
143	↓	175	—	207	Ï	239	ï
144	↑	176	°	208	Ð	240	ð
145	γ	177	±	209	Ñ	241	ñ
146	ó	178	²	210	Ò	242	ò
147	ε	179	³	211	Ó	243	ó
148	η	180	´	212	Ô	244	ô
149	θ	181	µ	213	Õ	245	õ
150	λ	182	¶	214	Ö	246	ö
151	ρ	183	·	215	×	247	×
152	σ	184	¸	216	Ø	248	ø
153	τ	185	¹	217	Ù	249	ù
154	ω	186	º	218	Ú	250	ú
155	△	187	»	219	Û	251	û
156	Π	188	¼	220	Ü	252	ü
157	Ω	189	½	221	Ý	253	ý
158	■	190	¾	222	ÿ	254	ÿ
159	∞	191	¿	223	ß	255	ÿ

Menünummern

Die folgende Tabelle enthält die Standardmenüs des HP 48 und die zugehörigen Menünummern.

Menü #	Menüname	Menü #	Menüname
0	Last Menu	19	I/O SETUP
1	CST	20	MODES
2	VAR	21	MODES Customization
3	MTH	22	MEMORY
4	MTH PARTS	23	MEMORY Arithmetic
5	MTH PROB	24	LIBRARY
6	MTH HYP	25	PORT 0
7	MTH MATR	26	PORT 1
8	MTH VECTR	27	PORT 2
9	MTH BASE	28	EDIT
10	PRG	29	SOLVE
11	PRG STK	30	SOLVE SOLVR
12	PRG OBJ	31	PLOT
13	PRG DISP	32	PLOT PTYPE
14	PRG CTRL	33	PLOT PLOTR
15	PRG BRCH	34	ALGEBRA
16	PRG TEST	35	TIME
17	PRINT	36	TIME ADJST
18	I/O	37	TIME ALRM

Menü #	Menüname	Menü #	Menüname
38	TIME ALRM RPT	49	UNITS FORCE
39	TIME SET	50	UNITS ENRG
40	STAT	51	UNITS POWR
41	STAT MODL	52	UNITS PRESS
42	UNITS Catalog	53	UNITS TEMP
43	UNITS LENG	54	UNITS ELEC
44	UNITS AREA	55	UNITS ANGL
45	UNITS VOL	56	UNITS LIGHT
46	UNITS TIME	57	UNITS RAD
47	UNITS SPEED	58	UNITS VISC
48	UNITS MASS	59	UNITS Command

Zusammenstellung der System-Flags des HP 48

Dieser Anhang enthält die System-Flags des HP 48 nach Funktionsgruppen geordnet. Alle Flags können gesetzt, rückgesetzt und getestet werden. Der Standardzustand der Flages ist *rückgesetzt*; davon ausgenommen sind nur die Flags, die die Wortlänge von Binärzahlen bestimmen (Flags -5 bis -10).

System-Flags

Flag	Beschreibung
Symbol-Flags für mathematische Funktionen	
-1	Hauptlösung. <i>Rückgesetzt:</i> QUAD und ISOL liefern ein Ergebnis, das alle möglichen Lösungen darstellt. <i>Gesetzt:</i> QUAD und ISOL liefern nur die Hauptlösung.
-2	Symbolische Darstellung von Konstanten. <i>Rückgesetzt:</i> Symbolische Konstanten (e, i, π , MAXR, und MINR) bleiben bei der Berechnung in ihrer symbolischen Form erhalten, wenn nicht das Flag -3 für die numerische Darstellung der Ergebnisse gesetzt ist. <i>Gesetzt:</i> Symbolische Konstanten werden durch ihren Zahlenwert ersetzt, unabhängig vom Zustand des Flag -3.

System-Flags (Fortsetzung)

Flag	Beschreibung
Symbol-Flags für mathematische Funktionen (Fortsetzung)	
-3	<p>Numerische Darstellung von Ergebnissen.</p> <p><i>Rückgesetzt:</i> Funktionen mit symbolischen Argumenten einschließlich symbolischer Konstanten werden nach der Berechnung in symbolischer Form dargestellt.</p> <p><i>Gesetzt:</i> Funktionen mit symbolischen Argumenten einschließlich symbolischer Konstanten werden nach der Berechnung numerisch dargestellt.</p>
-4	Wird nicht verwendet.
Binärzahlen-Flags	
-5 bis -10	<p>Wortlänge von Binärzahlen</p> <p>Mit den Flags -5 bis einschließlich -10 wird die Wortlänge zwischen 1 und 64 Bit eingestellt.</p>
-11 und -12	<p>Zahlenbasis</p> <p>HEX: -11 <i>gesetzt</i>, -12 <i>gesetzt</i>.</p> <p>DEC: -11 <i>rückgesetzt</i>, -12 <i>rückgesetzt</i>.</p> <p>OCT: -11 <i>gesetzt</i>, -12 <i>rückgesetzt</i>.</p> <p>BIN: -11 <i>rückgesetzt</i>, -12 <i>gesetzt</i>.</p>
-13 und -14	Wird nicht verwendet.
Koordinatensystem-Flags	
-15 und -16	<p>Rechteckkoordinaten: -15 <i>rückgesetzt</i>, -16 <i>rückgesetzt</i>.</p> <p>Polar-/Zylinderkoordinaten: -15 <i>rückgesetzt</i>, -16 <i>gesetzt</i>.</p> <p>Polar-/Kugelkoordinaten: -15 <i>gesetzt</i>, -16 <i>gesetzt</i>.</p>

System-Flags (Fortsetzung)

Flag	Beschreibung
Flags zur Behandlung mathematischer Sonderfälle (Forts.)	
-27 bis -29	Wird nicht verwendet.
Flags für die grafische Darstellung	
-30	<p>Grafische Darstellung von Funktionen.</p> <p><i>Rückgesetzt:</i> Bei Gleichungen der Form $y = f(x)$ wird nur $f(x)$ grafisch dargestellt.</p> <p><i>Gesetzt:</i> Bei Gleichungen der Form $y = f(x)$ werden y und $f(x)$ getrennt grafisch dargestellt.</p>
-31	<p>Verbinden von Punkten zu einer Kurve.</p> <p><i>Rückgesetzt:</i> Punkte werden miteinander zu einer Kurve verbunden.</p> <p><i>Gesetzt:</i> Punkte werden nicht verbunden.</p>
-32	<p>Grafikcursor.</p> <p><i>Rückgesetzt:</i> Der Grafikcursor ist immer dunkel.</p> <p><i>Gesetzt:</i> Der Grafikcursor erscheint dunkel auf hellem Hintergrund und hell auf dunklem Hintergrund.</p>
E/A- und Druck-Flags	
-33	<p>E/A-Gerät.</p> <p><i>Rückgesetzt:</i> Daten für nichtdruckendes E/A-Gerät werden zum seriellen Anschluß übertragen.</p> <p><i>Gesetzt:</i> Daten für nichtdruckendes E/A-Gerät werden zum IR-Anschluß übertragen.</p>
-34	<p>Drucker.</p> <p><i>Rückgesetzt:</i> Daten für den Drucker werden zum IR-Anschluß übertragen.</p> <p><i>Gesetzt:</i> Daten für den Drucker werden zum seriellen Anschluß übertragen.</p>

System-Flags (Fortsetzung)

Flag	Beschreibung
E/A- und Druck-Flags (Fortsetzung)	
- 35	<p>E/A-Datenformat.</p> <p><i>Rückgesetzt:</i> Objekte werden im ASCII-Format übertragen.</p> <p><i>Gesetzt:</i> Objekte werden im Speicherbildformat übertragen.</p>
- 36	<p>Überschreiben mit RECV.</p> <p><i>Rückgesetzt:</i> Wenn ein vom HP 48 empfangener Dateiname mit einem vorhandenen Variablennamen des HP 48 übereinstimmt, wird ein neuer Variablennamen mit einem Ziffernzusatz angelegt, um das Überschreiben der vorhandenen Variablen zu verhindern.</p> <p><i>Gesetzt:</i> Wenn ein vom HP 48 empfangener Dateiname mit einem vorhandenen Variablennamen des HP 48 übereinstimmt, wird die vorhandene Variable überschrieben.</p>
- 37	<p>Zweizeiliger Druck.</p> <p><i>Rückgesetzt:</i> Einzeiliger Druck.</p> <p><i>Gesetzt:</i> Zweizeiliger Druck.</p>
- 38	<p>Zeilenvorschub.</p> <p><i>Rückgesetzt:</i> Am Ende jeder Druckzeile wird das Zeichen für Zeilenvorschub zum Drucker geschickt.</p> <p><i>Gesetzt:</i> Das Zeichen für Zeilenvorschub wird am Zeilenende nicht geschickt.</p>
- 39	<p>E/A-Meldungen.</p> <p><i>Rückgesetzt:</i> E/A-Meldungen werden angezeigt.</p> <p><i>Gesetzt:</i> E/A-Meldungen werden unterdrückt.</p>
Zeit- und Datum-Flags	
- 40	<p>Anzeige der Uhr.</p> <p><i>Rückgesetzt:</i> Die Uhr wird nur angezeigt, wenn das Menü TIME gewählt wurde.</p> <p><i>Gesetzt:</i> Die Uhr wird immer angezeigt.</p>

System-Flags (Fortsetzung)

Flag	Beschreibung
Zeit- und Datum-Flags (Fortsetzung)	
- 41	<p>Zeitanzeigeformat. <i>Rückgesetzt:</i> 12-Stunden-Anzeige. <i>Gesetzt:</i> 24-Stunden-Anzeige.</p>
- 42	<p>Datumsformat. <i>Rückgesetzt:</i> Format MM/DD/YY (Monat/Tag/Jahr). <i>Gesetzt:</i> Format DD.MM.YY (Tag.Monat.Jahr).</p>
- 43	<p>Wiederholungstermine nicht neu ansetzen. <i>Rückgesetzt:</i> Nicht bestätigte Wiederholungstermine zu festen Zeiten werden automatisch neu angesetzt. <i>Gesetzt:</i> Nicht bestätigte Wiederholungsalarme zu festen Zeiten werden nicht neu angesetzt.</p>
- 44	<p>Bestätigte Alarmer werden gespeichert. <i>Rückgesetzt:</i> Bestätigte Alarmer werden aus der Terminliste gelöscht. <i>Gesetzt:</i> Bestätigte Termine werden weiter in der Terminliste gespeichert.</p>
Anzeigeformat-Flags	
- 45 bis - 48	<p>Anzahl der Dezimalstellen. Mit den Flags - 45 bis einschließlich - 48 wird die Anzahl der Dezimalstellen im FIX-, SCI- und ENG-Modus eingestellt.</p>
- 49 und - 50	<p>Zahlendarstellungsformat. FIX-Modus: - 49 <i>rückgesetzt</i>, - 50 <i>rückgesetzt</i>. Fix: - 49 <i>gesetzt</i>, - 50 <i>rückgesetzt</i>. SCI-Modus: - 49 <i>rückgesetzt</i>, - 50 <i>gesetzt</i>. ENG-Modus: - 49 <i>gesetzt</i>, - 50 <i>gesetzt</i>.</p>

System-Flags (Fortsetzung)

Flag	Beschreibung
Anzeigeformat-Flags (Fortsetzung)	
-51	<p>Dezimaltrennzeichen</p> <p><i>Rückgesetzt:</i> Das Dezimalzeichen ist der . (Punkt).</p> <p><i>Gesetzt:</i> Das Dezimalzeichen ist das , (Komma).</p>
-52	<p>Einzeilige Anzeige.</p> <p><i>Rückgesetzt:</i> Das Objekt in Ebene 1 wird angezeigt, wobei maximal vier Zeilen der Stackanzeige verwendet werden.</p> <p><i>Gesetzt:</i> Die Anzeige des Objektes in Ebene 1 ist auf eine Zeile beschränkt.</p>
-53	<p>Vorrang von Rechenoperationen.</p> <p><i>Rückgesetzt:</i> Bestimmte Klammern in algebraischen Ausdrücken werden unterdrückt, um die Lesbarkeit zu verbessern.</p> <p><i>Gesetzt:</i> Alle Klammern in algebraischen Ausdrücken werden dargestellt.</p>
-54	Wird nicht verwendet.
Sonstige Flags	
-55	<p>Letztes Argument.</p> <p><i>Rückgesetzt:</i> Die Argumente einer Operation werden gesichert.</p> <p><i>Gesetzt:</i> Die Argumente einer Operation werden nicht gesichert.</p>
-56	<p>Akustisches Fehlersignal.</p> <p><i>Rückgesetzt:</i> Die akustischen Signale bei einer Fehlermeldung und bei Ausführung des Befehls BEEP sind aktiviert.</p> <p><i>Gesetzt:</i> Die akustischen Signale bei einer Fehlermeldung und bei Ausführung des Befehls BEEP werden unterdrückt.</p>

System-Flags (Fortsetzung)

Flag	Beschreibung
Sonstige Flags (Fortsetzung)	
- 57	<p>Akustisches Alarmsignal.</p> <p><i>Rückgesetzt:</i> Das akustische Alarmsignal ist aktiviert.</p> <p><i>Gesetzt:</i> Das akustische Alarmsignal wird unterdrückt.</p>
- 58	<p>Ausführliche Meldungen.</p> <p><i>Rückgesetzt:</i> Meldungen mit Bedienerhinweisen und Daten werden automatisch angezeigt.</p> <p><i>Gesetzt:</i> Die automatische Anzeige von Meldungen mit Bedienerhinweisen und Daten wird unterdrückt.</p>
- 59	<p>Schnelle Kataloganzeige.</p> <p><i>Rückgesetzt:</i> Der Gleichungskatalog (und die Meldungen in den Menüs SOLVE, SOLVR, PLOT und PLOTR) zeigen die Gleichung und den Gleichungsnamen an.</p> <p><i>Gesetzt:</i> Der Gleichungskatalog (und die Meldungen in den Menüs SOLVE, SOLVR, PLOT, and PLOTR) zeigen nur den Gleichungsnamen an.</p>
- 60	<p>Verriegelung des Alpha-Modus.</p> <p><i>Rückgesetzt:</i> Der Alpha-Modus wird durch zweimaliges Drücken von  verriegelt.</p> <p><i>Gesetzt:</i> Der Alpha-Modus wird durch einmaliges Drücken von  verriegelt.</p>
- 61	<p>Verriegeln des Benutzermodus.</p> <p><i>Rückgesetzt:</i> Der Benutzermodus wird durch einmaliges Drücken von  [USR] verriegelt. Der Benutzermodus wird durch zweimaliges Drücken von  [USR] verriegelt.</p> <p><i>Gesetzt:</i> Der Benutzermodus wird durch einmaliges Drücken von  [USR] verriegelt.</p>
- 62	<p>Benutzermodus.</p> <p><i>Rückgesetzt:</i> Der Benutzermodus ist nicht aktiviert.</p> <p><i>Gesetzt:</i> Der Benutzermodus ist aktiviert.</p>

System-Flags (Fortsetzung)

Flag	Beschreibung
Sonstige Flags (Fortsetzung)	
-63	Benutzerdefiniertes ENTER . <i>Rückgesetzt:</i> ENTER wertet die Befehlszeile aus. <i>Gesetzt:</i> Das benutzerdefinierte ENTER ist aktiviert.
-64	Anzeiger für den Index. <i>Rückgesetzt:</i> Die letzte Ausführung von GETI oder PUTI hat den Index beim Erhöhen nicht auf den Anfangswert zurückgesetzt. <i>Gesetzt:</i> Die letzte Ausführung von GETI oder PUTI hat den Index beim Erhöhen auf den Anfangswert zurückgesetzt.

Verzeichnis der Operationen

Dieses Verzeichnis enthält alle Informationen, die für die Arbeit mit dem HP 48 erforderlich sind. Für jede Operation gibt dieses Verzeichnis an:

Name, Taste oder Menüfeld. Jeder Operation ist ein Name, eine Taste oder ein Menüfeld zugeordnet. Operationsnamen erscheinen als Tasten oder Menüfelder.

Beschreibung. Was die Operation bewirkt (oder welches ihr Wert ist, wenn es sich um eine Einheit handelt).

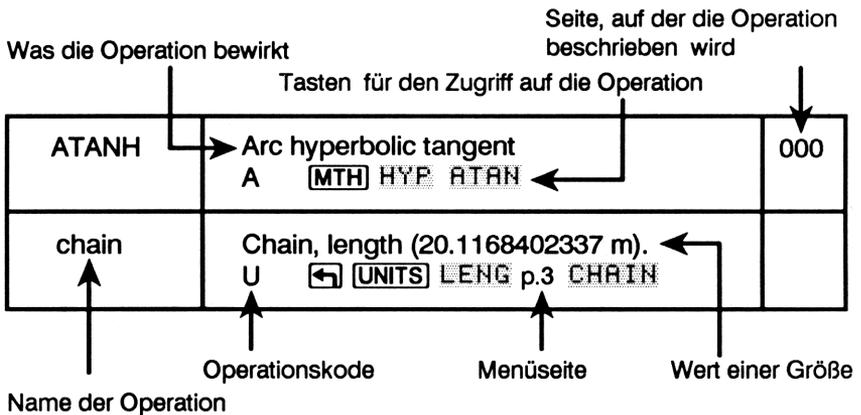
Art. Die Art der Operation wird durch einen der folgenden Buchstaben gekennzeichnet.

Op.-Kode	Beschreibung
O	Operation. Eine Operation, die nicht in die Befehlszeile, in ein Programm oder in einen algebraischen Ausdruck aufgenommen werden kann.
C	Befehl. Eine Operation, die in Programme, aber <i>nicht</i> in algebraische Ausdrücke aufgenommen werden kann.
F	Funktion. Eine Operation, die in algebraische Ausdrücke aufgenommen werden kann.
A	Analytische Funktion. Eine Funktion, für die der HP 48 die Umkehrfunktion und die Ableitung bilden kann.
U	Einheit.

Tasten. Die Tasten, über die man eine Operation aufrufen kann. Wenn einer Tastenfolge ein “...” vorangestellt ist, so ist der Zugriff über mehr als ein Menü möglich — die Tastenfolgen, die durch das “...” dargestellt werden, können Sie in diesem Verzeichnis der Beschreibung der Operation entnehmen, die unmittelbar auf das “...” folgt. Bei Operationen in mehrseitigen Menüs finden Sie auch die entsprechende Seitenzahl des Menüs. Operationen, die sich nicht über die Tastatur ausführen lassen, werden durch die “Bitte eintippen” gekennzeichnet.

Seite. Die Seite, auf der die Operation in diesem Handbuch beschrieben ist.

Die Eintragungen in diesem Verzeichnis sind in der folgenden Weise geordnet:



Operationen, deren Name alphabetische und Sonderzeichen enthalten, sind alphabetisch aufgelistet. Operationen, deren Namen ausschließlich Sonderzeichen enthalten, sind am Ende dieses Verzeichnisses zusammengestellt.

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
a	Ar, Fläche (100 m ²). U [UNITS] AREA p.2	
A	Ampere, elektrischer Strom (1 A). U [UNITS] p.2 ELEC	
Å	Angstrom, Länge (1 × 10 ⁻¹⁰ m). U [UNITS] LENG p.4	
R R	Verbinden links. O [EQUATION] RULES R Führt R aus, bis keine Änderung im Klammerausdruck mehr auftritt. O [EQUATION] RULES R	439 444
R R	Verbinden rechts. O [EQUATION] RULES R Führt R aus, bis keine Änderung im Klammerausdruck mehr auftritt. O [EQUATION] RULES R	439 444
ABS	Absolutwert. PARTS ABS MATR p.2 ABS F VECTR ABS	159
ACK	Bestätigt den angezeigten überfälligen Termin. C [TIME] ACK	483

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ACKALL	Bestätigt alle überfälligen Termine. C  TIME  ACKR	483
ACOS	Arcuscosinus. A  ACOS	150
ACOSH	Area cosinus hyperbolicus. A  MTH  HYP  ACOSH	147
acre	Acre, Fläche (4046,87260987 m ²). U  UNITS  RER p.2  ACRE	
 ADJUST	Wählt das Menü TIME ADJUST (einstellen). O  TIME  ADJUST	
 AF	Addiert Brüche. O  EQUATION  RULES  AF	443
  ALGEBRA	Wählt das Menü ALGEBRA. O  ALGEBRA	
  ALGEBRA	Wählt den Gleichungskatalog. O  ALGEBRA	280
ALOG	Exponentialfunktion (Basis 10). A  10 ^x	147

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ALRM	Wählt das Menü TIME ALRM (Alarm). O \leftarrow [TIME] ALRM	
AND	Logisches oder binäres UND. MTH BASE p.4 AND F [PRG] TEST AND	227 531
ANGL	Wählt das Menü UNITS ANGL . O \leftarrow [UNITS] p.3 ANGL	
APPLY	Übergibt ausgewertete Ausdrücke als Argument(e) an einen nicht ausgewerteten lokalen Namen. F \leftarrow [ALGEBRA] p.2 APPLY	
ARC	Erzeugt in PICT einen Kreisbogen mit dem Radius r und einem Mittelpunkt bei (x,y) von θ_1 nach θ_2 . C [PRG] DSPL ARC	366
ARCHIVE	Legt eine Sicherungskopie des Dateiverzeichnis HOME an. C \leftarrow [MEMORY] p.3 ARCHI	703
arcmin	Bogenminute, Winkel in der Ebene. $(4,62962962963 \times 10^{-5})$ U \leftarrow [UNITS] p.3 ANGL ARCFI	
arcs	Bogensekunde, Winkel in der Ebene. $(7,71604938272 \times 10^{-7})$ U \leftarrow [UNITS] p.3 ANGL ARCS	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
AREA	Berechnet und stellt die Fläche unter einem Funktionsgraphen zwischen zwei x-Werten dar, die durch die Marke und den Cursor definiert sind; überträgt die Fläche zurück in den Stack. O ... FCN AREA	332
AREA	Wählt das Menü UNITS AREA. O [←] [UNITS] AREA	
ARG	Liefert den Polarwinkel θ F [MTH] PARTS ARG	179
ARG	Aktiviert/sperrt die Möglichkeit, das letzte Argument mit LASTARG zurückzuholen. O [←] [MODES] p.2 ARG	240
ARRAY→	Übergibt Elemente eines Feldes an den Stack. C Bitte eintippen.	
→ARRAY	Setzt Einzelelemente zu einem Feld zusammen. C [PRG] [OBJ] →ARR	95
ASCII	Schaltet zwischen ASCII- und Binärmodus um. O [←] [I/O] SETUP ASCII	669
ASIN	Arcussinus. A [←] [ASIN]	150
ASINH	Area sinus hyperbolicus. A [MTH] [HYP] ASINH	147

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ASN	Belegt eine einzelne Taste mit einer benutzerspezifischen Definition. C  MODES  ASN 	236
ASR	Schiebt arithmetisch um ein Bit nach rechts. C MTH  BASE p.3  ASR 	227
ATAN	Arcustangens. A  ATAN 	150
ATANH	Area tangens hyperbolicus. A MTH  HYP  ATAN 	147
atm	Atmosphäre, Druck (101325 kg/m·s ²) U  UNITS p.2 PRESS  ATM 	
ATTACH	Bindet eine angegebene Bibliothek an das aktuelle Verzeichnis an. C  MEMORY p.2 ATTAC 	707
ATTN (ON)	Bricht die Ausführung eines Programms ab; löscht die Befehlszeile; verläßt bestimmte Menüumgebungen; löscht Meldungen. O ON 	58
AU	Astronomische Einheit, Länge (1,495979 × 10 ¹¹ m). U  UNITS  LENG p.2  AU 	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
AUTO	Skaliert automatisch die y-Achse. ... PLOTR AUTO C   AUTO	315
 AUTO	Skaliert automatisch die y-Achse; stellt danach die Funktion grafisch dar. ... PLOTR AUTO O   AUTO	318
AXES  AXES	Bestimmt die Koordinaten des Schnittpunktes der Koordinatenachsen; legt die Achsbeschriftung fest. ... PLOTR p.3 AXES C   p.3 AXES Legt die Koordinaten des Schnittpunkts der Koordinatenachsen im Stack ab. ... PLOTR p.3  AXES O   p.3  AXES	346 344
 A/PM	Schaltet die Uhr zwischen AM und PM um. O   SET  A/PM Schaltet die Uhr zwischen AM und PM um. O   ALRM  A/PM	477 479
b	Barn, Fläche($1 \times 10^{-28} \text{ m}^2$). U   AREA  B	
bar	Bar, Druck ($1 \times 10^5 \text{ Pa}$). U   p.2 PRESS  BAR	
BAR	Wählt den Plottyp BAR. C ... PTYPE BAR	354

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
BARPLOT	Erzeugt ein Balkendiagramm der Daten in ΣDAT . C [STAT] p.3 BARPL	409
BASE	Wählt das Menü MTH BASE. O [MTH] BASE	
BAUD	Stellt eine von vier möglichen Übertragungsraten ein. C [I/O] SETUP BAUD	669
bbl	Barrel, Volumen (0,158987294928 m ³). U [UNITS] VOL p.4 BBL	
BEEP	Erzeugt ein akustisches Signal. C [PRG] CTRL p.3 BEEP	561
BEEP	Aktiviert/sperrt das akustische Fehlersignal BEEP. O [MODES] BEEP	240
BESTFIT	Wählt das statistische Modell, das den größten Korrelationskoeffizienten (Absolutwert) liefert und führt LR aus. C [STAT] p.4 MODL BEST	405
BIN	Wählt für die Zahlendarstellung den Binärmodus. [MTH] BASE BIN C [MODES] p.4 BIN	225
BINS	Sortiert Werte der unabhängigen Variablen aus ΣDAT in $N + 2$ Intervalle. (bis maximal 1048573 Intervalle). C [STAT] p.2 BINS	411

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
BLANK	Legt ein leeres grafisches Objekt an. C [PRG] [DSPL] p.3 [BLAN]	370
BOX	Erzeugt ein Rechteck, dessen gegenüberliegende Ecken durch angegebene Koordinaten bestimmt sind. C [PRG] [DSPL] [BOX]	366
[BOX]	Zeichnet ein Rechteck, dessen gegenüberliegende Ecken durch die Marke und den Cursor definiert sind. ... [DRAW] p.2 [BOX] ... [AUTO] p.2 [BOX] O [↩] [GRAPH] p.2 [BOX]	364
Bq	Becquerel, Aktivität(1 1/s). U [↩] [UNITS] p.3 [RAD] [BQ]	
[BRCH]	Wählt das Menü PRG BRCH (Programmzweig). O [PRG] [BRCH]	
Btu	Internationales Btu, Energie (1055,05585262 kg·m ² /s ²) U [↩] [UNITS] p.2 [ENRG] [BTU]	
bu	Bushel, Volumen (0,03523907 m ³). U [↩] [UNITS] [VOL] p.4 [BU]	
BUFLEN	Liefert die Anzahl der Zeichen im seriellen Puffer. C [↩] [I/O] p.3 [BUFLE]	687

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
BYTES	Liefert die Größe eines Objekts (in Bytes) und seine Prüfsumme. C  MEMORY BYTES	107
B→R	Wandelt eine Binärzahl in eine reelle Zahl um. C MTH BASE p.2 B→R	228
c	Lichtgeschwindigkeit (299792458 m/s). U  UNITS SPEED p.2 C	
C	Coulomb, elektrische Ladung (1 A·s). U  UNITS p.2 ELEC C	
°C	Grad Celsius, Temperatur. U  UNITS p.2 TEMP °C	
cal	Kalorie, Energie (4,186 kg·m ² /s ²) U  UNITS p.2 ENRG CAL	
CASE	Leitet eine CASE-Struktur ein. C PRG BRCH CASE	537
 CASE	Erzeugt ein CASE THEN END END als Schreibhilfe. O PRG BRCH  CASE	537
 CASE	Erzeugt ein THEN END als Schreibhilfe. O PRG BRCH  CASE	537

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
<p>CAT</p>	<p>Wählt den Gleichungskatalog.</p> <p>← PLOT CAT</p> <p>← SOLVE CAT</p> <p>○ → ALGEBRA</p> <p>Wählt den Katalog STAT.</p> <p>○ ← STAT CAT</p> <p>Wählt den Terminkatalog.</p> <p>← TIME CAT</p> <p>○ → TIME</p>	<p>280</p> <p>398</p> <p>482</p>
<p>cd</p>	<p>Candela, Lichtstärke (1 cd).</p> <p>U ← UNITS p.3 LIGHT p.2 CD</p>	
<p>CEIL</p>	<p>Liefert die nächstgrößere ganze Zahl.</p> <p>F MTH PARTS p.3 CEIL</p>	<p>159</p>
<p>CENT</p>	<p>Plottet die grafische Darstellung neu, mit dem Mittelpunkt der Darstellung an der aktuellen Cursorposition.</p> <p>... DRAW CENT</p> <p>... AUTO CENT</p> <p>○ ← GRAPH CENT</p>	<p>316</p>

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
CIRCL	Erzeugt einen Kreis mit dem Mittelpunkt an der Marke und einem Radius gleich dem Abstand zwischen Cursor und Marke. ... DRAW p.2 CIRCL ... AUTO p.2 CIRCL O [←] GRAPH p.2 CIRCL	364
CKSM	Wählt eines von drei möglichen Prüfsummenverfahren zur Fehlererkennung. C [←] I/O SETUP CKSM	670
CLEAR	Löscht den Stack. C [→] CLR	68
[CLR]	Löscht das Display im Eingabemodus des EquationWriter. O [←] EQUATION [→] CLR Löscht PICT. ... DRAW [→] CLR ... AUTO [→] CLR O [←] GRAPH [→] CLR	248 327
CLK	Schaltet die Anzeige der Uhr ein und aus. O [←] MODES p.2 CLK	240

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
CLKADJ	Addiert eine bestimmte Anzahl von Uhrzyklen zur Systemzeit. C  TIME ADJUST p.2 CLKR	476
CLLCD	Blendet die Stackanzeige aus. C PRG DSPL p.4 CLLCD	562
CLOSEIO	Schließt den E/A-Anschluß. C  I/O p.2 CLOSE	667
CLΣ	Löscht die statistischen Daten in ΣDAT. C  STAT CLΣ	395
CLUSR	Löscht alle Benutzervariablen. C Bitte eintippen.	
CLVAR	Löscht alle Benutzervariablen. C  PURGE	123
cm	Zentimeter, Länge (0,01 m). U  UNITS LENG CM	
CMD	Aktiviert/sperrt die Möglichkeit die letzte Befehlszeile zurückzuholen. O  MODES p.2 CMD	240
cm ²	Quadratcentimeter, Fläche ($1 \times 10^{-4} \text{ m}^2$). U  UNITS AREA CM^2	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
cm ³	Kubikzentimeter, Volumen ($1 \times 10^{-6} \text{ m}^3$). U  [UNITS] VOL CM ³	
cm/s	Zentimeter pro Sekunde, Geschwindigkeit (0,01 m/s). U  [UNITS] SPEED CM/S	
CNCT	Schaltet das Verbinden von Punkten zu einem geschlossenen Kurvenzug ein und aus. O  [MODES] p.2 CNCT	240
CNRM	Berechnet die Spaltennorm einer Matrix (größte Spaltensumme). C [MTH] MATR p.2 CNRM	386
+COL	Fügt im MatrixWriter an der Cursorposition eine Zeile von Nullen ein. O  [MATRIX] p.2 +COL	377
-COL	Löscht die aktuelle Spalte im MatrixWriter. O  [MATRIX] p.2 -COL	377
COLCT	Faßt gleiche Terme in einem algebraischen Ausdruck zusammen. C  [ALGEBRA] COLCT	425
COLCT	Faßt gleiche Glieder in einem angegebenen Klammerausdruck zusammen. O  [EQUATION]  RULES COLCT	432
COLΣ	Legt die abhängige und die unabhängige Spalte in ΣDAT fest. C Bitte eintippen.	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
COMB	Liefert die Zahl der Kombinationen von n aus m Elementen. F [MTH] PROB COMB	158
CON	Erzeugt eine Matrix mit Konstanten. C [MTH] MATR CON	386
CONIC	Wählt den Plottyp CONIC. C ... PTYPE CONIC	351
CONJ	Liefert die Konjugierte einer komplexen Zahl. F [MTH] PARTS CONJ	179
CONT	Setzt ein unterbrochenes Programm fort. C [↩] CONT	564
CONVERT	Wandelt die Einheit eines Objektes in eine andere Einheit um. C [↗] UNITS CONV	210
COORD	Zeigt die Cursorkoordinaten links unten im Display an. ... DRAW COORD ... AUTO COORD O [↩] GRAPH COORD	326

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
CORR	Berechnet den Korrelationskoeffizienten der statistischen Daten in Σ DAT. C  [STAT] p.4 	405
COS	Cosinus. A 	150
COSH	Cosinus hyperbolicus. A   	147
COV	Berechnet die Kovarianz der statistischen Daten in Σ DAT. C  [STAT] p.4 	405
CR	Bewirkt einen Druckkopfrücklauf/Zeilenvorschub des Druckers. C  [PRINT] 	653
CRDIR	Legt ein Verzeichnis an. C  [MEMORY] 	129
CROSS	Kreuzprodukt von Vektoren mit zwei oder drei Elementen. C   	379
	Wählt das (Benutzer-)Menü CST. O 	230
	Liefert den Inhalt der Variablen CST. O  [MODES] 	231

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ct	Karat, Masse (0,0002 kg). U  [UNITS] MASS p.2 CT	
	Wählt das Menü PRG CTRL (Programmsteuerung). O [PRG] CTRL	
cu	US cup, Volumen ($2,365882365 \times 10^{-4}$ m ³). U  [UNITS] VOL p.3 CU	
C→PX	Wandelt Benutzerkoordinaten in Pixelkoordinaten um. C [PRG] DSPL p.2 C→PX	348
C→R	Zerlegt eine komplexe Zahl in zwei reelle Zahlen. C [PRG] OBJ pg.2 C→R	96
d	Tag, Zeit (86400 s). U  [UNITS] TIME D	
	Zusammensetzen oder Zerlegen einer komplexen Zahl oder eines 2-dimensionalen Vektors. O  [2D]	173
	Zusammensetzen oder Zerlegen eines 3-dimensionalen Vektors. O  [3D]	186

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
←D	Verteilen nach links. O [←] EQUATION [←] RULES ←D	436
[→] ←D	Führt ←D aus, bis keine Änderung mehr im Klammerausdruck auftritt. O [←] EQUATION [←] RULES [→] ←D	440
D→	Verteilen nach rechts. O [←] EQUATION [←] RULES D→	436
[→] D→	Führt D→, aus, bis keine Änderung mehr im Klammerausdruck auftritt. O [←] EQUATION [←] RULES [→] D→	440
DATE	Liefert das Systemdatum. C [←] TIME p.2 DATE	489
DATE +	Berechnet ein neues Datum aus einem angegebenen Datum und einer Anzahl von Tagen. C [←] TIME p.2 DATE+	488
→DATE	Stellt ein Datum für einen Termin ein. C [←] TIME SET →DAT	474
>DATE	Stellt ein Datum für einen Termin ein. O [←] TIME ALRM DATE	478
DAY	Stellt das Terminwiederholungsintervall auf <i>n</i> Tage ein. O [←] TIME ALRM RPT DAY	478

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
[DEBUG]	Unterbricht die Programmausführung vor dem ersten Objekt. O [PRG] [CTRL] [DEBUG]	521
DDAYS	Berechnet die Anzahl von Tagen zwischen zwei Daten. C [←] [TIME] p.2 [DDAYS]	489
DEC	Wählt für die Zahlendarstellung den Dezimalmodus. [MTH] [BASE] [DEC] C [←] [MODES] p.4 [DEC]	225
DECR	Verringert den Wert einer angegebenen Variablen um 1. C [→] [MEMORY] [DECR]	553
DEFINE	Erzeugt eine Variable oder eine benutzerdefinierte Funktion. C [←] [DEF]	115 162
→DEF	Entwickelt trigonometrische und hyperbolische Funktionen in Ausdrücken, die EXP und LN enthalten. O [←] [EQUATION] [←] [RULES] →DEF	440
DEG	Wählt den Winkelmodus Grad. C [←] [MODES] p.3 [DEG]	150
[DEL]	Löscht das Zeichen unter dem Cursor. O [DEL]	81

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
DEL	<p>Löscht eine rechteckige Fläche, deren gegenüberliegende Ecken durch die Marke und den Cursor definiert sind.</p> <p>... DRAW p.3 DEL</p> <p>... AUTO p.3 DEL</p> <p>O [←] GRAPH p.3 DEL</p>	362
<p>←DEL</p> <p>[→] ←DEL</p>	<p>Löscht alle Zeichen vom Cursor bis zum Beginn des Wortes.</p> <p>[←] EDIT ←DEL</p> <p>O ... EDIT ←DEL</p> <p>Löscht alle Zeichen vom Cursor bis zum Zeilenanfang.</p> <p>[←] EDIT [→] ←DEL</p> <p>O ... EDIT [→] ←DEL</p>	73
<p>DEL→</p> <p>[→] DEL→</p>	<p>Löscht alle Zeichen vom Cursor bis zum Anfang des nächsten Wortes.</p> <p>[←] EDIT DEL→</p> <p>O ... EDIT DEL→</p> <p>Löscht alle Zeichen vom Cursor bis zum Zeilenende.</p> <p>[←] EDIT [→] DEL→</p> <p>O ... EDIT [→] DEL→</p>	73
DELALARM	<p>Löscht einen Termin aus der Terminliste.</p> <p>C [←] TIME ALRM p.2 DELAL</p>	487

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
DELAY	Stellt die Verzögerungszeit zwischen zwei Zeilen, die an den Drucker gesendet werden, ein. C  PRINT p.2 DELAY	653
DELKEYS	Löscht benutzerdefinierte Tastenbelegungen. C  MODES DELK	236
DEPND  DEPN	Gibt die abhängige Plotvariable an. ... PLOTR p.2 DEPN C  PLOT p.2 DEPN Legt die abhängige Plotvariable im Stack ab. ... PLOTR p.2  DEPN O  PLOT p.2  DEPN	342 342
DEPTH	Legt die Anzahl der Datenblöcke im Stack ab. C PRG STK DEPTH	84
DET	Berechnet die Determinante einer Matrix. C MTH MATR DET	387
DETACH	Koppelt eine angegebene Bibliothek vom aktuellen Verzeichnis ab. C  MEMORY p.2 DETAC	709

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
DINV	Invertiert doppelt. O  [EQUATION]  RULES DINV	431
DISP	Stellt ein Objekt in einer angegebenen Zeile des Anzeigefeldes dar. C [PRG] DSPL p.4 DISP C [PRG] CTRL p.2 DISP	561
DNEG	Negiert doppelt. O  [EQUATION]  RULES DNEG	430
DO  DO	Beginnt eine unbestimmte Schleife. C [PRG] BRCH DO Erzeugt DO UNTIL END als Schreibhilfe. O [PRG] BRCH  DO	549 549
DOERR	Bricht die Programmausführung ab und stellt eine angegebene Meldung im Display dar. C [PRG] CTRL p.3 DOERR	592
DOT	Skalarprodukt von zwei Vektoren. C [MTH] VECTR DOT	380

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
DOT+	Schaltet die Bildpunkte ein, über die sich der Cursor bewegt. ... DRAW p.2 DOT+ ... AUTO p.2 DOT+ O  GRAPH p.2 DOT+	364
DOT-	Schaltet die Bildpunkte aus, über die sich der Cursor bewegt. ... DRAW p.2 DOT- ... AUTO p.2 DOT- O  GRAPH p.2 DOT-	364
DRAW	Stellt eine Gleichung grafisch ohne Koordinatenachsen dar. ... PLOTR DRAW C  PLOT DRAW	314
DRAW	Stellt eine Gleichung grafisch mit Koordinatenachsen dar. ... PLOTR DRAW O  PLOT DRAW	319
DRAX	Stellt die Koordinatenachsen dar. ... PLOTR p.3 DRAX C  PLOT p.3 DRAX	343

Name, Taste oder Menüfeld	Bezeichnung Art, Tasten	Seite
DROP	Entfernt ein Objekt aus Ebene 1; verschiebt alle übrigen Objekte um eine Ebene nach unten. C  DROP	69
DROPN	Entfernt n Objekte aus dem Stack. C PRG STK pg.2 DRPN	84
DRPN	Entfernt alle Objekte aus dem Stack vom Zeiger bis zur Ebene 1. O ... STK p.2 DRPN	76
DROP2	Entfernt die beiden ersten Objekte vom Stack. C PRG STK pg.2 DROP2	84
DSPL	Wählt das Menü PRG DSPL (Programmanzeige) O PRG DSPL	
DTAG	Löscht alle Markierungen eines Objektes. C PRG OBJ pg.2 DTAG	98
DUP	Dupliziert das Objekt in Ebene 1. C PRG STK pg.2 DUP	70
DUPN	Dupliziert n Objekte im Stack. C PRG STK pg.2 DUPN	84
DUPN	Dupliziert alle Datenblöcke im Stack vom Zeiger bis einschließlich Ebene 1. O ... STK p.2 DUPN	76

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
DUP2	Dupliziert Objekte in Ebene 1 und Ebene 2. C [PRG] [STK] pg.2 DUP2	84
dyn	dyn, Kraft (0,00001 kg·m/s ²). U [↵] [UNITS] p.2 FORCE DYN	
D→R	Umwandlung von Grad in Radiant. F [MTH] VECTR p.2 D→R	153
e	Eulersche Zahl e (2,71828182846). A [α] [↵] [E]	155
ECHO	Kopiert das Objekt in der aktuellen Ebene in die Befehlszeile. O ... [↑STK] ECHO	76
EDEQ	Übergibt den Inhalt von EQ zum Editieren in die Befehlszeile. [↵] [PLOT] EDEQ O [↵] [SOLVE] EDEQ	275

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
<p>EDIT</p>	<p>Wenn die Befehlszeile nicht aktiv ist, wird das Objekt aus Ebene 1 in die Befehlszeile kopiert und das Menü EDIT gewählt.</p> <p>Wenn die Befehlszeile bereits aktiv ist, wird das Menü EDIT gewählt.</p> <p>○ ← EDIT</p> <p>Wählt das Menü EDIT.</p> <p>○ → MATRIX ← EDIT</p> <p>Übergibt die Gleichung in die Befehlszeile und wählt das Menü EDIT.</p> <p>○ ← EQUATION ← EDIT</p> <p>Editiert die aktuelle Stackebene.</p> <p>○ ... ↑STK ← EDIT</p>	<p>71</p> <p>73</p> <p>376</p> <p>260</p> <p>77</p>
<p>EDIT</p>	<p>Kopiert die gewählte Gleichung in die Befehlszeile und wählt das Menü EDIT.</p> <p>← PLOT CAT EDIT</p> <p>← SOLVE CAT EDIT</p> <p>○ → ALGEBRA EDIT</p> <p>Kopiert einen Teilausdruck in die Befehlszeile und wählt das Menü EDIT.</p> <p>○ ← EQUATION ← EDIT</p> <p>Kopiert die gewählte Matrix in den MatrixWriter.</p> <p>○ ← STAT CAT EDIT</p> <p>Editiert das aktuelle Matrixelement.</p> <p>○ → MATRIX EDIT</p> <p>Zeigt den gewählten Termin an und wählt das Menü ALRM (Alarm).</p> <p>○ ← TIME CAT EDIT</p>	<p>279</p> <p>262</p> <p>399</p> <p>377</p> <p>484</p>

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
EDITΣ	Kopiert statistische Daten von ΣDAT zum MatrixWriter. O  STAT EDITΣ	396
EEX	Gibt E ein oder bewegt den Cursor zu einem vorhandenen Exponenten in der Befehlszeile. O  EEX	51
 ELEC	Wählt das Menü UNITS ELEC (elektrische Einheiten). O  UNITS p.2 ELEC	
erg	erg, Energie (0,0000001 kg·m ² ;s ²) U  UNITS p.2 ENRG ENRG	
ELSE	Leitet eine ELSE-Anweisung ein. C  PRG BRCH p.3 ELSE	535
END	Beendet Programmstrukturen. C  PRG BRCH p.3 END	533
ENG	Stellt den ENG-Modus für die Anzeige ein. C  MODES ENG	62
 ENRG	Wählt das Menü UNITS ENRG (Energie). O  UNITS p.2 ENRG	
 ENTER	Gibt den Inhalt einer Befehlszeile in den Stack ein. Ist keine Befehlszeile vorhanden, wird DUP ausgeführt. O  ENTER	106

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
[ENTRY]	Schaltet zwischen dem algebraischen Eingabemodus und dem Programmeingabemodus um. O [→] [ENTRY]	83
[EQUATION]	Aktiviert den EquationWriter. O [←] [EQUATION]	
EQ+ [←] EQ+	Fügt die ausgewählte Gleichung zur Liste in EQ hinzu. [←] [PLOT] CAT EQ+ [←] [SOLVE] CAT EQ+ O [→] [ALGEBRA] EQ+ Entfernt den letzten Eintrag aus der Liste in EQ. [←] [PLOT] CAT [←] EQ+ [←] [SOLVE] CAT [←] EQ+ O [→] [ALGEBRA] [←] EQ+	293 293
EQ→	Teilt eine Gleichung in eine linke und eine rechte Seite auf. C [PRG] [OBJ] EQ→	98
ERASE	Löscht PICT. ... PLOTR ERASE C [→] [PLOT] ERASE	314

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ERRM	Gibt die letzte Fehlermeldung aus. C [PRG] CTRL p.3 ERRM	588
ERRN	Gibt die letzte Fehlernummer aus. C [PRG] CTRL p.3 ERRN	588
ERRO	Löscht die letzte Fehlernummer. C [PRG] CTRL p.3 ERRO	588
eV	Elektronenvolt, Energie ($1,60219 \times 10^{-19} \text{ kg} \cdot \text{m}^2 / \text{s}^2$) U [←] [UNITS] p.2 ENRG p.2 EV	
EVAL	Wertet ein Objekt aus. C [EVAL]	105
EXEC	Legt die Wirkung eines Termins fest. O [←] [TIME] ALRM EXEC Legt das Objekt, das an einem Termin ausgeführt wird, im Stack ab. O [←] [TIME] ALRM [→] EXEC	478 478
EXECS	Zeigt an, was an einem Termin geschieht. [←] [TIME] CAT EXECS O [→] [TIME] EXECS	484

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
EXIT	Verläßt die Auswahlumgebung. O \leftarrow EQUATION \leftarrow EXIT Verläßt das Menü FCN (Funktion). O ... FCN EXIT Verläßt das Menü ZOOM. O ... ZOOM EXIT	429 331 328
EXP	Exponentialfunktion (Basis e). A \leftarrow e^x	147
EXPAN	Erweitert einen algebraischen Datenblock. C \leftarrow ALGEBRA EXP	426
EXPFIT	Wählt die Exponentialfunktion als Näherungsmodell. C \leftarrow STAT p.4 MODL EXPR	405
EXPM	Exponentialfunktion (Basis e) minus 1 ($e^x - 1$). A MTH HYP p.2 EXPM	147
EXPR	Hebt den Teilausdruck hervor, für den das angegebene Objekt die Hauptfunktion ist. O \leftarrow EQUATION \leftarrow EXPR	266 428
EXPR=	Gibt Werte von algebraischen Ausdrücken oder Gleichungen aus. O ... SOLVR EXPR=	285
EXTR	Bewegt den grafischen Cursor zum nächsten Extremwert, zeigt die Koordinaten an und legt sie im Stack ab. O ... FCN EXTR	331

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
E^{\wedge}	Ersetzt ein Produkt zweier Potenzen durch die Potenz einer Potenz. O \leftarrow [EQUATION] \leftarrow RULES E^{\wedge}	439
$E(\)$	Ersetzt die Potenz einer Potenz durch ein Produkt zweier Potenzen. O \leftarrow [EQUATION] \leftarrow RULES $E(\)$	408
F	Farad, Kapazität ($1 \text{ A}^2 \cdot \text{s}^4 / \text{kg} \cdot \text{m}^2$). U \leftarrow [UNITS] p.2 ELEC F	
°F	Grad Fahrenheit, Temperatur. U \leftarrow [UNITS] p.2 TEMP °F	
FRST	Schaltet um zwischen der Darstellung nur von Gleichungsnamen und der Darstellung der Namen zusammen mit den Gleichungen. O ... CAT p.2 FRST	280
fath	Fathom, Länge (1,82880365761 m). U \leftarrow [UNITS] LENG p.3 FATH	
fbm	Board foot, Volumen (0,002359737216 m^3). U \leftarrow [UNITS] VOL p.4 FBM	
fc	Footcandle, Beleuchtungsstärke (0,856564774909 cd/m^2) U \leftarrow [UNITS] p.3 LIGHT FC	
FCN	Wählt das Menü GRAPHICS FCN (Funktion). ... DRAW FCN ... AUTO FCN O \leftarrow [GRAPH] FCN	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
FC?	Testet, ob eine bestimmtes Flag rückgesetzt ist. <code>[PRG] TEST p.3 FC?</code> C <code>[→] [MODES] p.3 FC?</code>	556
FC?C	Testet, ob eine bestimmtes Flag rückgesetzt ist; setzt es anschließend zurück. <code>[PRG] TEST p.3 FC?C</code> C <code>[→] [MODES] p.3 FC?C</code>	556
Fdy	Farad, elektrische Ladung (96487 A·s). U <code>[←] [UNITS] p.2 ELEC p.2 FDY</code>	
fermi	Fermi, Länge (1×10^{-15} m). U <code>[←] [UNITS] LENG p.4 FERMI</code>	
FINDALARM	Zeigt den ersten Termin, der nach einem angegebenen Zeitpunkt fällig wird, an. C <code>[←] [TIME] ALRM p.2 FINDA</code>	488
FINISH	Beendet den Kermit-Servermodus. C <code>[←] [/O] FINIS</code>	667
FIX	Stellt den FIX-Modus für die Anzeige ein. C <code>[←] [MODES] FIX</code>	62
flam	Foot lambert, Leuchtdichte (3,42625909964 cd/m ²) U <code>[←] [UNITS] p.3 LIGHT FLAM</code>	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
FLOOR	Nächstkleinere ganze Zahl. F [MTH] PARTS p.3 FLOOR	159
FM,	Schaltet zwischen den Dezimalzeichen Punkt und Komma um. O [←] MODES p.4 FM,	62
FOR [←] FOR [→] FOR	Beginnt eine bestimmte Schleife. C [PRG] BRCH FOR Erzeugt FOR NEXT als Schreibhilfe. O [PRG] BRCH [←] FOR Erzeugt FOR STEP als Schreibhilfe. O [PRG] BRCH [→] FOR	545 545 547
FORCE	Wählt das Menü UNITS FORCE. O [←] [UNITS] p.2 FORCE	
FP	Liefert den Nachkommateil einer Zahl. F [MTH] PARTS p.3 FP	159
FREE	Ersetzt ein Objekt im RAM durch eine neue Kopie des Objektes. C [←] [MEMORY] p.3 FREE	704
FREEZE	"Friert" einen oder mehrere von drei Anzeigebereichen ein. C [PRG] DSPL p.4 FREEZ	370 562

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
FS?	Testet, ob ein bestimmtes Flag gesetzt ist. PRG TEST p.3 FS? C ↩ MODES p.3 FS?	556
FS?C	Testet, ob eine bestimmtes Flag gesetzt ist; setzt es anschließend zurück. PRG TEST p.3 FS?C C ↩ MODES p.3 FS?C	556
ft	International foot, Länge (0,3048 m). U ↩ UNITS LENG FT	
ft ²	Square foot, Fläche (0,09290304 m ²). U ↩ UNITS AREA FT^2	
ft ³	Cubic foot, Volumen (0,028316846592 m ³). U ↩ UNITS VOL FT^3	
ftUS	Survey foot, Länge (0,304800609601 m). U ↩ UNITS LENG p.3 FTUS	
ft/s	Feet/second, Geschwindigkeit (0,3048 m/s). U ↩ UNITS SPEED FT/S	
ft*lbf	Foot-poundf, Energie (1,35581794833 kg·m ² /s ²). U ↩ UNITS p.2 ENRG FT*LB	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
FUNCTION	Wählt den Plottyp FUNCTION. C ... PTYPE FUNC	352
F(X)	Zeigt den Wert der Funktion für den durch die Cursorposition angegebenen x-Wert. Legt den Funktionswert im Stack ab. O ... FCN p.2 F(X)	331
F'	Stellt die erste Ableitung der Funktion sowie die Funktion selbst dar und fügt die Ableitung EQ hinzu. O ... FCN p.2 F'	332
g	Gramm, Masse (0,001 kg). U [↩] UNITS MASS G	
ga	Normfallbeschleunigung, Beschleunigung (9,80665 m/s ²). U [↩] UNITS SPEED p.2 GA	
gal	US gallon, Volumen (0,003785411784 m ³). U [↩] UNITS VOL p.2 GAL	
galC	Canadian gallon, Volumen (0,004546092m ³). U [↩] UNITS VOL p.2 GALC	
galUK	UK gallon, Volumen (0,004546092 m ³). U [↩] UNITS VOL p.2 GALU	
GET	Holt ein Element aus einem Feld oder einer Liste. C [PRG] OBJ p.4 GET	98

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
GETI	Holt ein Element aus einem Feld oder einer Liste und erhöht den Index. C PRG OBJ p.4 GETI	99
gf	Gramm, Kraft (0,00980665 kg·m/s ²). U UNITS p.2 FORCE GF	
GOR	Überlagert zwei Grafikobjekte. C PRG DSPL p.3 GOR	369
GO↓	Wählt den Eingabemodus "Von oben nach unten". O MATRIX GO↓	377
GO→	Wählt den Eingabemodus "Von links nach rechts". O MATRIX GO→	377
GRAD	Wählt den Winkelmodus GON. C MODES p.3 GRAD	149
grad	Grad, Winkel in der Ebene (0,0025). U UNITS p.3 ANGL GRAD	
grain	Grain, Masse (0,00006479891 kg). U UNITS MASS p.2 GRAIN	
GRAPH	Ruft die Grafikumgebung auf. C GRAPH	324

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
[GRAPH]	Ruft den Rollmodus auf. [←] [EQUATION] [←] [GRAPH] ... DRAW [←] [GRAPH] ... AUTO [←] [GRAPH] O [←] [GRAPH] [←] [GRAPH]	246 326
→GROB	Konvertiert ein Objekt in ein Grafikobjekt. C [PRG] DSPL p.3 →GRO	368
GXOR	Überlagert ein Grafikobjekt einem invertierten Grafikobjekt. C [PRG] DSPL p.3 GXOR	369
Gy	Gray, absorbierte Dosis ($1 \text{ m}^2/\text{s}^2$). U [←] [UNITS] p.3 RAD GY	
h	Stunde, Zeit (3600 s). U [←] [UNITS] TIME H	
H	Henry, Induktivität ($1 \text{ kg}\cdot\text{m}^2/\text{A}^2\cdot\text{s}^2$). U [←] [UNITS] p.2 ELEC p.2 H	
*H	Stellt den vertikalen Plotmaßstab ein. C ... PLOTR p.3 *H	343
ha	Hektar, Fläche (10000 m^2). U [←] [UNITS] AREA p.2 HA	
HALT	Hält die Programmausführung an. C [PRG] CTRL HALT	522 561

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
HEX	Wählt für die Zahlendarstellung den Hexadezimalmodus. [MTH] BASE HEX C [←] [MODES] p.4 HEX	224
HISTPLOT	Erzeugt das Histogramm der Daten in Σ DAT. C [←] [STAT] p.3 HISTP	407
HISTOGRAM	Wählt den Plottyp HISTOGRAM. C ... PTYPE p.2 HIST	353
HMS+	Addiert im HMS-Format. C [←] [TIME] p.3 HMS+	153 492
HMS-	Subtrahiert im HMS-Format. C [←] [TIME] p.3 HMS-	153 492
HMS→	Konvertiert vom HMS- in das Dezimalformat. C [←] [TIME] p.3 HMS→	153 491
→HMS	Konvertiert vom Dezimal- in das HMS-Format. C [←] [TIME] p.3 →HMS	491 491
HOME	Wählt das HOME-Verzeichnis. C [→] [HOME]	131
HOURL	Stellt das Wiederholintervall für Termine auf n Stunden ein. O [←] [TIME] ALRM RPT HOURL	479
hp	Horsepower, Leistung (745,699871582 kg·m ² /s ³). U [←] [UNITS] p.2 POWR HP	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
HR+	Stellt die Uhr eine Stunde vor. O \leftarrow [TIME] ADJUST HR+	476
HR-	Stellt die Uhr eine Stunde zurück. O \leftarrow [TIME] ADJUST HR-	476
HYP	Wählt das Menü MTH HYP (Mathematik - Hyperbelfunktionen). O [MTH] HYP	
Hz	Hertz, Frequenz (1/s). U \leftarrow [UNITS] TIME HZ	
i	Imaginäre Einheit i . A α \leftarrow [CST]	155
IDN	Legt eine Einheitsmatrix bestimmter Größe an. C [MTH] MATR IDN	387
IF	Beginnt einen Testausdruck. C [PRG] BRCH IF	532
\leftarrow IF	Erzeugt IF THEN END als Schreibhilfe. O [PRG] BRCH \leftarrow IF	532
\rightarrow IF	Erzeugt IF THEN ELSE END als Schreibhilfe. O [PRG] BRCH \rightarrow IF	535

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
IFERR	Beginnt einen Testausdruck. C [PRG] BRCH p.3 IFERR	588
[←] IFERR	Erzeugt IFERR THEN END als Schreibhilfe.	588
[→] IFERR	O [PRG] BRCH p.3 [←] IFERR Erzeugt IFERR THEN ELSE END als Schreibhilfe. O [PRG] BRCH p.3 [→] IFERR	590
IFT	Befehl IF-THEN. C [PRG] BRCH p.3 IFT	539
IFTE	Funktion IF-THEN-ELSE. F [PRG] BRCH p.3 IFTE	539
IM	Liefert den Imaginärteil einer komplexen Zahl oder eines komplexen Feldes. F [MTH] PARTS IM	179
in	Inch, Länge (0,0254 m). U [←] [UNITS] LENG IN	
in ²	Square inch, Fläche (0,00064516 m ²). U [←] [UNITS] AREA IN ²	
in ³	Cubic inch, Volumen (0,000016387064 m ³). U [←] [UNITS] VOL IN ³	

Name, Taste oder Menüfeld	Bezeichnung Art, Tasten	Seite
INCR	Erhöht den Wert einer angegebenen Variablen um 1. C   INCR	553
INDEP	Gibt die unabhängige Plotvariable an. ... PLOTR INDEP C   INDEP Holt die unabhängige Variable in den Stack zurück. ... PLOTR  INDEP O    INDEP	316 315
inHg	Inch Quecksilbersäule, Druck (3386,38815789 kg/m-s ²). U   p.2 PRESS p.2 INHG	
inH2O	Inches of water, Druck (248,84 kg/m-s ²). U   p.2 PRESS p.2 INH2O	
INPUT	Unterbricht die Programmausführung, gibt eine Meldung aus und wartet auf Daten. C  CTRL p.2 INPUT	566
INS	Schaltet um zwischen Einfügen/Überschreiben von Zeichen. O   INS	73

Name, Taste oder Menüfeld	Bezeichnung Art, Tasten	Seite
INV	Kehrwert. A $\boxed{1/x}$	65
IP	Ganzzahliger Teil einer reellen Zahl. F \boxed{MTH} PARTS p.3 \boxed{IP}	159
$\boxed{IR/W}$	Schaltet um zwischen der Übertragung über den IR-Anschluß und der Übertragung über den seriellen Anschluß. O $\boxed{\leftarrow} \boxed{I/O}$ SETUP $\boxed{IR/W}$	669
\boxed{ISECT}	Bewegt den Grafikkursor zum nächsten Schnittpunkt von zwei Funktionen, zeigt die Schnittpunktkoordinaten an und legt sie im Stack ab. O ... \boxed{FCN} \boxed{ISECT}	331
ISOL	Isoliert eine Variable auf einer Seite einer Gleichung. C $\boxed{\leftarrow} \boxed{ALGEBRA}$ \boxed{ISOL}	421
$\boxed{\leftarrow} \boxed{I/O}$ $\boxed{\rightarrow} \boxed{I/O}$	Wählt das Menü I/O (Eingabe/Ausgabe). O $\boxed{\leftarrow} \boxed{I/O}$ Bringt den HP 48 in den Kermit-Servermodus. O $\boxed{\rightarrow} \boxed{I/O}$	677
J	Joule, Energie ($1 \text{ kg} \cdot \text{m}^2/\text{s}^2$). U $\boxed{\leftarrow} \boxed{UNITS}$ p.2 ENRG \boxed{J}	
K	Kelvin, Temperatur (1 K). U $\boxed{\leftarrow} \boxed{UNITS}$ p.2 TEMP \boxed{K}	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
kcal	Kilokalorie, Energie (4186 kg·m ² /s ²) U  UNITS p.2 ENRG KCAL	
KEEP	Löscht alle Ebenen oberhalb der aktuellen Ebene. O ...  STK p.2 KEEP	77
KERRM	Liefert den Text des zuletzt empfangenen KERMIT-Fehlerpaketes. C  I/O p.2 KERR	666
KEY	Liefert eine Nummer, die die letzte gedrückte Taste kennzeichnet. C  CTRL p.2 KEY	584
KEYS	Entfernt die Menüfelder aus dem Display um mehr Platz für die Grafik zu schaffen. ... DRAW p.3 KEYS ... AUTO p.3 KEYS O  GRAPH p.3 KEYS	327
kg	Kilogramm, Masse (1 kg). U  UNITS MASS KG	
KGET	Kermit-Befehl, holt Daten von einem anderen Gerät. C  I/O KGET	666
KILL	Bricht alle unterbrochenen Programme endgültig ab. C  CTRL KILL	522

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
kip	Kilopound, Kraft (4448,22161526 kg·m/s ²). U  UNITS p.2 FORCE KIP	
km	Kilometer, Länge (1 km). U  UNITS LENG p.2 KM	
km ²	Quadratkilometer, Fläche (1 km ²). U  UNITS AREA p.2 KM ²	
knot	Nautische Meilen pro Stunde, Geschwindigkeit (0,514444444444 m/s). U  UNITS SPEED KNOT	
kph	Kilometer pro Stunde, Geschwindigkeit (0,277777777778 m/s). U  UNITS SPEED KPH	
l	Liter, Volumen (0,001 m ³). U  UNITS VOL p.2 L	
LABEL	Beschriftet Achsen. ... PLOTR p.3 LABEL C  PLOT p.3 LABEL	344

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
LABEL	Beschriftet Achsen. ... DRAW LABEL ... AUTO LABEL O  GRAPH LABEL	326
lam	Lambert, Leuchtdichte (3183,09886184 cd/m ²). U  UNITS p.3 LIGHT p.2 LAM	
LAST	Holt letzte(s) Argument(e) zurück in den Stack. C Bitte eintippen.	
LASTARG	Holt letzte(s) Argument(e) zurück in den Stack. C  LAST ARG	69
 LAST CMD	Zeigt den vorhergehenden Inhalt der Befehlszeile an. O  LAST CMD	83
 LAST MENU	Wählt die zuletzt angezeigte Seite des vorgehenden Menüs. O  LAST MENU	61
 LAST STACK	Stellt den vorhergehenden Zustand des Stacks wieder her. O  LAST STACK	80

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
lb	Avoirdupois pound, Masse (0,45359237 kg). U  [UNITS] MASS LB	
lbf	Gewichtskraft (4,44822161526 kg·m/s ²). U  [UNITS] p.2 FORCE LBF	
lbt	Troy pound, Masse (0,3732417 kg). U  [UNITS] MASS LBT	
LCD→	Legt die gegenwärtige Stackanzeige als Grafikobjekt im Stack ab. C [PRG] DSPL p.4 LCD→	371
→LCD	Stellt das angegebene Grafikobjekt in der Stackanzeige dar. C [PRG] DSPL p.4 →LCD	371
LENG	Wählt das Menü UNITS LENG (Länge). O  [UNITS] LENG	
LEVEL	Überträgt die Nummer der aktuellen Ebene in Ebene 1. O ... ↑STK p.2 LEVEL	77
[LIBRARY]	Wählt das Menü LIBRARY. O  [LIBRARY]	

Name, Taste oder Menüfeld	Bezeichnung Art, Tasten	Seite
LIBS	Listet alle Bibliotheken auf, die an das aktuelle Verzeichnis angebunden sind. C \leftarrow [MEMORY] p.2 LIBS	709
LIGHT	Wählt das Menü UNITS LIGHT. O \leftarrow [UNITS] p.3 LIGHT	
LINE	Zieht eine Linie zwischen den Punkten, deren Koordinaten in den Ebenen 1 und 2 stehen. C [PRG] DSPL LINE	366
LINE	Zieht eine Linie von der Marke zum Cursor. ... DRAW p.2 LINE ... AUTO p.2 LINE O \leftarrow [GRAPH] p.2 LINE	364
Σ LINE	Liefert die bestangepaßte Kurve für die Daten in Σ DAT entsprechend dem gewählten statistischen Näherungsmodell zurück. C \leftarrow [STAT] p.3 Σ LINE	407
LINFIT	Wählt die Gerade als Näherungsmodell. C \leftarrow [STAT] p.4 MODL LIN	408
LIST \rightarrow	Legt die Elemente einer Liste im Stack ab. C Bitte eingeben.	
\rightarrow LIST	Faßt angegebene Objekte in einer Liste zusammen. C [PRG] OBJ \rightarrow LIST	99
\rightarrow LIST	Faßt Objekte von Ebene 1 bis einschließlich der aktuellen Ebene in einer Liste zusammen. O ... \uparrow STK \rightarrow LIST	76

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
lm	Lumen, Lichtstrom ($7,95774715459 \times 10^{-2}$ cd). U  UNITS p.3 LIGHT LM	
LN	Natürlicher Logarithmus (Basis e). A  LN	65
LNP1	Natürlicher Logarithmus von (Argument + 1). A MTH HYP p.2 LNP1	148
LOG	Zehnerlogarithmus (Basis 10). A  LOG	147
LOGFIT	Wählt die Logarithmusfunktion als Näherungsmodell. C  STAT p.4 MODL LOG	406
LR	Berechnet die lineare Regression. C  STAT p.4 LR	407
lx	Lux, Beleuchtungsstärke ($7,95774715459 \times 10^{-2}$ cd/m ²). U  UNITS p.3 LIGHT LX	
lyr	Light year, Länge ($9,46052840488 \times 10^{15}$ m). U  UNITS LENG p.2 LYR	
L*	Ersetzt den Logarithmus einer Potenz durch das Produkt mit einem Logarithmus. O  EQUATION  RULES L*	442

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
L<>	Ersetzt ein Produkt mit einem Logarithmus durch den Logarithmus einer Potenz. O [↩] EQUATION [◀] RULES L<>	442
←M [▶] ←M	Faßt Faktoren links zusammen. O [↩] EQUATION [◀] RULES ←M Führt ←M aus, bis keine Änderung mehr im Teilausdruck auftritt. O [↩] EQUATION [◀] RULES [▶] ←M	440 444
M→ [▶] M→	Faßt Faktoren rechts zusammen. O [↩] EQUATION [◀] RULES M→ Führt M→ aus, bis keine Änderung mehr im Klammerausdruck auftritt. O [↩] EQUATION [◀] RULES [▶] M→	441 444
m	Meter, Länge (1 m). U [↩] UNITS LENG M	
m ²	Quadratmeter, Fläche (1 m ²). U [↩] UNITS AREA M ²	
m ³	Kubikmeter, Volumen (1 m ³). U [↩] UNITS VOL M ³	
MANT	Mantisse einer Zahl. F [MTH] PARTS p.3 MANT	159

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
MARK	Setzt die Marke an die Cursorposition. ... DRAW p.3 MARK ... AUTO p.3 MARK O  GRAPH p.3 MARK	325
MASS	Wählt das Menü UNITS MASS. O  UNITS MASS	
↑MATCH	Muster suchen und ersetzen; beginnend mit den Teilausdrücken. C  ALGEBRA p.2 ↑MAT	449
↓MATCH	Muster suchen und ersetzen; beginnend mit dem Gesamtobjekt. C  ALGEBRA p.2 ↓MAT	449
MATR	Wählt das Menü MTH MATR (Mathematik - Matrizen). O  MATR	
MATRIX	Wählt den MatrixWriter. O  MATRIX	
MAX	Maximum von zwei reellen Zahlen. F  PARTS p.2 MAX	159

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
MAXR	Größte reelle Zahl, die mit dem Taschenrechner darstellbar ist. (9,999999999999E499). A [MTH] [PARTS] p.4 MAXR	155
MAXΣ	Größte Werte in jeder Spalte in der statistischen Matrix in ΣDAT. C [←] [STAT] p.2 MAXΣ	404
MEAN	Berechnet den Mittelwert der Daten in ΣDAT. C [←] [STAT] p.2 MEAN	404
MEM	Anzahl der verfügbaren Speicherplätze (Bytes). C [←] [MEMORY] MEM	109
[←] [MEMORY] [→] [MEMORY]	Wählt das Menü MEMORY. O [←] [MEMORY] Wählt das Menü für Speicherarithmetik. O [→] [MEMORY]	133
MENU	Zeigt ein eingebautes oder das benutzerspezifische Menü an. [→] [MODES] MENU C [PRG] CTRL p.2 MENU	229 578

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
MERGE	Bindet den Speicher auf der RAM-Steckkarte in den Hauptspeicher ein. C [↩] [MEMORY] p.3 MERG	698
μ	Mikron, Länge (1×10^{-6} m). U [↩] [UNITS] LENG p.4 μ	
MeV	Megaelektronenvolt, Energie ($1,60219 \times 10^{-13}$ kg·m ² /s ²). U [↩] [UNITS] p.2 ENRG p.2 MEV	
mho	Siemens, elektrischer Leitwert (1 A ² ·s ³ /kg·m ²). U [↩] [UNITS] p.2 ELEC p.2 MHO	
mi	Internationale Meile, Länge (1609,344 m). U [↩] [UNITS] LENG p.2 MI	
mi ²	Internationale Quadratmeile, Fläche (2589988,11034 m ²). U [↩] [UNITS] AREA p.2 MI ²	
mil	Tausendstel Zoll, Länge (0,0000254 m). U [↩] [UNITS] LENG p.4 MIL	
min	Minute, Zeit (60 s). U [↩] [UNITS] TIME MIN	
MIN	Kleinster Wert von zwei reellen Zahlen. F [MTH] PARTS p.2 MIN	159

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
MIN	Stellt das Wiederholintervall für Termine auf n Minuten ein. O ← TIME ALRM RPT MIN	478
MINR	Kleinste reelle Zahl, die mit dem Taschenrechner darstellbar ist. ($1,000000000000E^{-499}$). A [MTH] PARTS p.4 MINR	155
MIN+	Stellt die Uhr um eine Minute vor. O ← TIME ADJST MIN+	476
MIN-	Stellt die Uhr um eine Minute zurück. O ← TIME ADJST MIN-	476
MINΣ	Kleinste Werte in jeder Spalte der statistischen Matrix in ΣDAT . C ← STAT p.2 MINΣ	404
miUS	US-Meile, Länge (1609,34721869 m). U ← UNITS LENG p.3 MIUS	
miUS²	US-Quadratmeile, Fläche (258998,47032 m ²). U ← UNITS AREA p.2 MIUS²	
mm	Millimeter, Länge (0,001 m). U ← UNITS LENG MM	
mmHg	Millimeter Quecksilbersäule (Torr), Druck (133,322368421 kg/m·s ²). U ← UNITS p.2 PRESS MMH	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ml	Milliliter (Kubikzentimeter), Volumen $(1 \times 10^{-6} \text{ m}^3)$. U  UNITS VOL p.3 ML	
 ML	Schaltet zwischen mehrzeiliger und einzeiliger Anzeige um. O  MODES p.2 ML	238
MOD	Modulo. F  MTH PARTS p.2 MOD	159
 MODES  MODES	Wählt das Menü MODES. O  MODES Wählt das Menü für die benutzerspezifische Tastaturbelegung. O  MODES	257 251
 MODL	Wählt das Menü STAT MODL (statistisches Modell). O  STAT p.4 MODL	408
mol	Mol, Masse (1 mol). U  UNITS MASS p.3 MOL	
Mpc	Megaparsec, Länge $(3,08567818585 \times 10^{22} \text{ m})$. U  UNITS LENG p.2 MPC	
mph	Meilen pro Stunde, Geschwindigkeit $(0,44704 \text{ m/s})$. U  UNITS SPEED MPH	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
MTH	Wählt das Menü MTH (Mathematik). O MTH	
M/D	Schaltet das Format der Datumsanzeige um. O ← TIME SET M/D	476
m/s	Meter pro Sekunde, Geschwindigkeit (1 m/s). U ← UNITS SPEED M/S	
N	Newton, Kraft (1 kg·m/s ²). U ← UNITS p.2 FORCE N	
NΣ	Anzahl der Zeilen in ΣDAT . C ← STAT p.5 NΣ	414
NEG	Negieren. A +/-	144
NEW	Holt ein algebraisches Objekt oder eine Matrix vom Stack, verlangt einen Namen und speichert das algebraische Objekt unter diesem Namen in EQ oder die Matrix in ΣDAT . ← PLOT NEW ← SOLVE NEW O ← STAT NEW	276 396
NEWOB	Trennt ein Objekt von einer Liste oder einem Variablennamen. C ← MEMORY p.2 NEWO	
NEXT	Beendet eine bestimmte Schleife. C PRG BRCH p.2 NEXT	540 544

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
NEXT	Zeigt das nächste oder die beiden nächsten Objekte in einem unterbrochenen Programm an, führt aber keine Operation mit ihnen aus. O [PRG] CTRL NEXT	521
nmi	Nautische Meile, Länge (1852 m). U [←] UNITS LENG p.3 NMI	
NONE	Löscht das Wiederholintervall für Termine und kehrt zum Menü TIME ALRM zurück. O [←] TIME ALRM RPT NONE	481
NOT	Logisches oder binäres NICHT. [PRG] TEST NOT F [MTH] BASE p.4 NOT	531 226
NUM	Liefert den Zeichenkode des ersten Zeichens in einer Zeichenkette. C [PRG] OBJ p.3 NUM	99
→NUM	Berechnet den Zahlenwert eines algebraischen Ausdrucks. C [→] →NUM	136
NXEQ	Läßt die Liste der Gleichungen in EQ schrittweise rotieren. ... SOLVR NXEQ O ... FCN p.2 NXEQ	292 332
NXT	Wählt die nächste Menüseite. O [NXT]	60

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
[OBJ]	Wählt das Menü PRG OBJ (Programmdatenblock). O [PRG] [OBJ]	
OBJ→	Legt Komponenten eines Objektes im Stack ab. C [PRG] [OBJ] [OBJ→]	100
OCT	Wählt für die Zahlendarstellung den Oktalmodus. [MTH] BASE [OCT] C [←] MODES p.4 [OCT]	223
[OFF]	Schaltet den Taschenrechner aus. O [→] [OFF]	25
OFF	Schaltet den Taschenrechner aus. C [PRG] CTRL p.3 [OFF]	580
OLDPRT	Führt eine Anpassung des HP 48-Zeichensatzes an den Infrarotdrucker HP 82240A durch. C [←] [PRINT] p.2 [OLDPR]	654
[ON]	Schaltet den Taschenrechner ein. O [ON]	25

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
OPENIO	Öffnet den seriellen Anschluß. C   p.2 OPENI	667
OR	Logisches oder binäres ODER.  BASE p.4  F  TEST 	226 531
ORDER	Ordnet das aktuelle Verzeichnis neu, so daß die Variablen im MenüUSR (Benutzer) in der Reihenfolge erscheinen, in der sie in einer Liste stehen. C   ORDER	121
	Setzt eine ausgewählte Gleichung an die erste Stelle der Liste im Gleichungskatalog.   CAT p.2 ORDER   CAT p.2 ORDER O   p.2 ORDER Setzt ausgewählte statistische Daten an die erste Stelle der Liste im Statistik-katalog. O   CAT p.2 ORDER	279 401
OVER	Kopiert ein Objekt von Ebene 2 nach Ebene 1. C  STK 	84
oz	Unze, Masse (0,028349523135 kg). U   MASS 	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ozfl	US Flüssigkeitsunze, Volumen (2,95735295625 × 10 ⁻⁵ m ³). U  UNITS VOL p.3 OZFL	
ozt	Unze (tr), Masse (0,031103475 kg). U  UNITS MASS p.2 OZT	
ozUK	UK Flüssigkeitsunze, Volumen (2,8413075 × 10 ⁻⁵ m ³). U  UNITS VOL p.3 OZUK	
P	Poise, dynamische Viskosität (0,1 kg/m·s) U  UNITS p.3 VISC P	
Pa	Pascal, Druck (1 kg/m·s ²) U  UNITS p.2 PRESS PA	
PARAMETRIC	Wählt den Plottyp PARAMETRIC. C ... PTYPE PARR	353
PARITY	Wählt eine von 5 möglichen Paritätseinstellungen. C  I/O SETUP PARIT	669
PARTS	Wählt das Menü MTH PARTS. O MTH PARTS	
PATH	Liefert eine Liste mit dem aktuellen Verzeichnispfad. C  MEMORY PATH	128

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
pc	Parsec, Länge ($3,08567818585 \times 10^{16}$ m). U  UNITS LENG p.2 PC	
PDIM  PDIM	Ändert die Größe von <i>PICT</i> PLOTR p.3 PDIM C  PLOT p.3 PDIM Legt die Größe von <i>PICT</i> im Stack ab. ... PLOTR p.3  PDIM O  PLOT p.3  PDIM	351 345
pdl	Poundal, Kraft ($0,138254954376$ kg·m/s ²). U  UNITS p.2 FORCE PDL	
PERM	Permutationen. F MTH PROB PERM	158
PGDIR	Löscht ein bestimmtes Verzeichnis. C  MEMORY p.3 PGDIR	132
ph	Phot, Beleuchtungsstärke ($795,774715459$ cd/m ²) U  UNITS p.3 LIGHT PH	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
PICK	Kopiert Objekt in Ebene <i>n</i> nach Ebene 1. C [PRG] [STK] PICK	84
PICK	Kopiert Objekt in der aktuellen Ebene nach Ebene 1. O ... [STK] PICK	76
PICT	Legt <i>PICT</i> in Ebene 1 ab. C [PRG] [DSPL] PICT	368
PIXOFF	Schaltet angegebenen Bildpunkt in <i>PICT</i> aus. C [PRG] [DSPL] p.2 PIXOF	367
PIXON	Schaltet angegebenen Bildpunkt in <i>PICT</i> ein. C [PRG] [DSPL] p.2 PIXON	367
PIX?	Prüft, ob ein angegebener Bildpunkt in <i>PICT</i> ein- oder ausgeschaltet ist. C [PRG] [DSPL] p.2 PIX?	367
pk	Peck, Volumen (0,0088097675 m ³). U [←] [UNITS] VOL p.4 PK	
PKT	Sendet KERMIT-Befehle an einen Server. C [←] [I/O] p.2 PKT	667

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
PLOT	Wählt das Menü PLOT. O ← PLOT Wählt das Menü PLOT PLOTR. O → PLOT	
PLOT	Macht den gewählten Eintrag zur aktuellen statistischen Matrix und zeigt die dritte Seite des Menüs STAT an. O ← STAT CAT PLOT	399
PLOTR	Wählt das Menü PLOT PLOTR. ← PLOT PLOTR ← PLOT CAT PLOTR → ALGEBRA PLOTR O ← SOLVE CAT PLOTR	
PMAX	Legt die Koordinaten der rechten oberen Ecke des Plotbereichs fest. C Bitte eintippen.	
PMIN	Legt die Koordinaten der linken unteren Ecke des Plotbereichs fest. C Bitte eintippen.	
POLAR	Schaltet zwischen Rechteck- und Polarkoordinaten um. O → POLAR	170
POLAR	Wählt den Plottyp POLAR. C ... PTYPE POLAR	351

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
POS	Liefert die Position einer Teilzeichenkette innerhalb einer Zeichenkette oder eines Objekts innerhalb einer Liste. C [PRG] [OBJ] p.3 [POS]	101
[POWER]	Wählt das Menü UNITS POWER (Leistung). O [←] [UNITS] p.2 [POWER]	
PREDEV	Vorhersagewert. C Bitte eingeben.	
PREDX	Berechnet einen Vorhersagewert für eine unabhängige Variable aus dem angegebenen Wert für eine abhängige Variable. C [←] [STAT] p.4 [PREDX]	407
PREDY	Berechnet einen Vorhersagewert für eine abhängige Variable aus dem angegebenen Wert für eine unabhängige Variable. C [←] [STAT] p.4 [PREDY]	407
[PRESS]	Wählt das Menü UNITS PRESS (Druck). O [←] [UNITS] p.2 [PRES]	
[←] [PREV]	Wählt die vorhergehende Menüseite. O [←] [PREV]	60
[→] [PREV]	Wählt die erste Menüseite. O [→] [PREV]	60

Name, Taste oder Menüfeld	Bezeichnung Art, Tasten	Seite
[PRG]	Wählt das Menü PRG (Programm). O [PRG]	
[PRINT]	Wählt das Menü PRINT. O [↩][PRINT]	
PRLCD	Druckt den Inhalt des Displays. C [↩][PRINT] PRLCD O Drücken Sie gleichzeitig [ON] [MTH]	653
[PROB]	Wählt das Menü MATH PROB (Wahrscheinlichkeit). O [MTH] [PROB]	
PROMPT	Zeigt eine Aufforderung (Zeichenkette) im Statusbereich an und unterbricht die Programmausführung. C [PRG] [CTRL] p.2 [PROM]	560
PRST	Druckt alle Objekte im Stack. C [↩][PRINT] [PRST]	653
PRSTC	Druckt alle Objekte im Stack im kompakten Format. C [↩][PRINT] [PRSTC]	653

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
PRVAR	Druckt Namen und Inhalt von einer oder mehreren Variablen (einschließlich Portnamen). C  PRINT PRVAR	653
PR1	Druckt Objekt in Ebene 1.  PRINT PR1 C  PRINT	653
psi	Pfund pro Quadratinch, Druck (6894,75729317 kg/m ²). U  UNITS p.2 PRESS PSI	
pt	pint, Volumen (0,000473176473 m ³). U  UNITS VOL p.2 PT	
P TYPE	Wählt das Menü PLOT PTYPE.  PLOT PTYPE ... PLOTR p.2 PTYPE O  PLOT p.2 PTYPE	
PURGE	Löscht eine oder mehrere angegebene Variablen. C  PURGE	122
PURGE	Löscht eine oder mehrere angegebene Variablen. Ist nur eine nichtmarkierte Variable angegeben, so wird ihr Inhalt für eine Wiederherstellung mit LASTARG gesichert. O  PURGE	122

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
PURG	<p>Löscht die gewählte Gleichung.</p> <p> <input type="button" value="←"/> SOLVE CAT p.2 PURG <input type="button" value="→"/> PLOT CAT p.2 PURG O <input type="button" value="→"/> ALGEBRA CAT p.2 PURG </p> <p>Löscht die gewählte statistische Matrix.</p> <p>O <input type="button" value="←"/> STAT CAT p.2 PURG</p> <p>Löscht den gewählten Termin.</p> <p> <input type="button" value="←"/> TIME CAT PURG O <input type="button" value="→"/> TIME PURG </p>	<p>279</p> <p>400</p> <p>484</p>
PUT	<p>Ersetzt ein Element in einem Feld oder einer Liste.</p> <p>C <input type="button" value="PRG"/> OBJ p.4 PUT</p>	99
PUTI	<p>Ersetzt ein Element in einem Feld oder einer Liste und erhöht den Index.</p> <p>C <input type="button" value="PRG"/> OBJ p.4 PUTI</p>	99
PVARs	<p>Liefert eine Liste der aktuellen Sicherungsobjekte und Bibliotheken in einem Port.</p> <p>C <input type="button" value="←"/> MEMORY p.2 PVARs</p>	702
PVIEW	<p>Zeigt <i>PICT</i> an, und zwar so, daß ein angegebener Bildpunkt sich in der linken oberen Ecke des Anzeigefeldes befindet.</p> <p>C <input type="button" value="PRG"/> DSPL PVIEW</p>	368

Name, Taste oder Menüfeld	Beschreibung Art, Taste	Seite
PWRFIT	Wählt die Potenzfunktion als Näherungsmodell. C [STAT] p.4 MODL PWR	405
PX→C	Wandelt Bildpunktkoordinaten in Benutzerkoordinaten um. C [PRG] DSPL p.2 PX→C	348
→Q	Wandelt eine Zahl in einen Näherungsbruch um. C	144
QUAD	Bestimmt die Lösungen von Polynomen erster und zweiter Ordnung. C [ALGEBRA] QUAD	420
QUOTE	Legt einen Ausdruck unausgewertet im Stack ab. F [ALGEBRA] p.2 QUOT	
qt	quart, Volumen (0,000946352946 m ³). U [UNITS] VOL p.2 QT	
→Q π	Berechnet und vergleicht Näherungen einer Zahl und der Zahl/ π durch einen Näherungsbruch. C [ALGEBRA] p.2 →Q π	144
r	Radian, Winkel in der Ebene (0,1591549343092). U [UNITS] p.3 ANGL R	
R	Roentgen, Strahlungsbelastung (0,000258 A·s/kg). U [UNITS] p.3 RAD p.2 R	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
°R	Grad Rankine, Temperatur. U [UNITS] p.2 °R	
rad	Rad, Energiedosis (0,01 m ² /s ²). U [UNITS] p.3	
RAD	Wählt den Winkelmodus Radiant. C [MODES] p.3	149
	Schaltet zwischen den Winkelmodi Radiant und Grad um. O [RAD]	32
	Wählt das Menü UNITS RAD (Strahlung). O [UNITS] p.3	
RAND	Liefert eine Zufallszahl. C	158
RATIO	Präfixform von /, wird durch EquationWriter verwendet. F Bitte eintippen.	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
RCEQ	Legt Gleichung aus <i>EQ</i> in Ebene 1 ab. ⬅️ PLOT ➡️ STEQ ⬅️ SOLVE ➡️ STEQ ... PLOTR ➡️ DRAW C ➡️ PLOT ➡️ DRAW	275
RCL	Legt ein in einer angegebenen Variablen gespeichertes Objekt im Stack ab. C ➡️ RCL	118
RCL	Fügt einen algebraischen Ausdruck von Ebene 1 in eine EquationWriter-Gleichung ein. O ➡️ RCL	264
RCLALARM	Ruft einen angegebenen Termin aus der Systemterminliste auf. C ⬅️ TIME ALRM p.2 RCLAL	487
RCLF	Liefert zwei Binärzahlen, die die Zustände der System- und der Benutzerflags darstellen. C ➡️ MODES p.2 RCLF	559
RCLKEYS	Liefert eine Liste der aktuellen benutzerdefinierten Tastenbelegungen. O ➡️ MODES RCLK	236
RCLMENU	Liefert die Menünummer des zuletzt aufgerufenen Menüs. C ➡️ MODES p.2 RCLM	579

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
RCLΣ	Legt die aktuelle statistische Matrix aus ΣDAT im Stack ab. C  STAT  STOΣ	396
RCWS	Liefert die aktuelle Wortlänge von Binärzahlen. C  BASE RCWS	222
rd	rod, Länge (5,0292100584 m). U  UNITS LENG p.3 RD	
RDM	Dimensionsiert ein Feld neu. C  MATR RDM	387
RDZ	Legt den Ausgangswert für die Erzeugung von Zufallszahlen fest. C  PROB RDZ	158
RE	Liefert den Realteil einer komplexen Zahl oder eines Feldes. F  PARTS RE	178
RECN	Wartet auf Daten von einem anderen Gerät, das mit dem Kermit-Protokoll arbeitet. Die Daten erhalten einen Namen, der im Stack angegeben ist. C  I/O p.2 RECN	667
RECV	Wartet auf Daten von einem anderen Gerät, das mit dem Kermit-Protokoll arbeitet. Die Daten behalten den vom Sender angegebenen Namen. C  I/O RECV	666

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
rem	rem, Dosisäquivalent (0,01 m ² /s ²). U  UNITS p.3 RAD REM	
REPEAT	Beginnt die REPEAT Anweisung. C PRG BRCH p.2 REPEA	552
REPL	Ersetzt den Teil eines Objektes durch ein anderes passendes Objekt. PRG OBJ p.3 REPL C PRG DSPL p.3 REPL	100 369
REPL	Ersetzt einen Teil von <i>PICT</i> durch ein Grafikobjekt in Ebene 1. ... DRAW p.3 REPL ... AUTO p.3 REPL O  GRAPH p.3 REPL	369
REPL	Ersetzt einen bestimmten Teilausdruck durch einen algebraischen Ausdruck vom Stack. O  EQUATION  REPL	266 428

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
<p>RES</p> <p>☞ RES</p>	<p>Stellt den Abstand zwischen den Stützpunkten einer grafischen Darstellung ein.</p> <p>... PLOTR p.2 RES</p> <p>C ☞ [PLOT] p.2 RES</p> <p>Legt den aktuellen Abstand im Stack ab.</p> <p>... PLOTR p.2 ☞ RES</p> <p>O ☞ [PLOT] p.2 ☞ RES</p>	<p>345</p> <p>342</p>
<p>RESET</p>	<p>Setzt die Plotparameter in <i>PPAR</i> im aktuellen Verzeichnis auf ihre Standardwerte zurück, löscht <i>PICT</i> und stellt seine Größe neu ein.</p> <p>... PLOTR p.2 RESET</p> <p>O ☞ [PLOT] p.2 RESET</p>	<p>347</p>
<p>RESTORE</p>	<p>Ersetzt das Verzeichnis <i>HOME</i> durch eine Sicherungskopie.</p> <p>C ☞ [MEMORY] p.3 RESTO</p>	<p>703</p>

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
REVIEW	<p>Zeigt die statistischen Daten in ΣDAT an. <input type="radio"/> <input type="button" value="←"/> STAT <input type="button" value="←"/> REVIEW</p> <p>Zeigt die aktuelle Gleichung und die Plotparameter an. ... PLOTR <input type="button" value="←"/> REVIEW <input type="button" value="→"/> PLOT <input type="button" value="←"/> REVIEW ... DRAW <input type="button" value="←"/> REVIEW ... AUTO <input type="button" value="←"/> REVIEW</p> <p><input type="radio"/> <input type="button" value="←"/> GRAPH <input type="button" value="←"/> REVIEW</p> <p>Zeigt die aktuelle Gleichung an. <input type="radio"/> <input type="button" value="←"/> SOLVE <input type="button" value="←"/> REVIEW <input type="button" value="←"/> PLOT <input type="button" value="←"/> REVIEW</p> <p>Zeigt die aktuelle Gleichung und die Werte der SOLVR-Variablen an. <input type="radio"/> ... SOLVR <input type="button" value="←"/> REVIEW</p> <p>Stellt Einheitenamen entsprechend dem gewählten Menü dar. <input type="radio"/> <input type="button" value="←"/> UNITS ... <input type="button" value="←"/> REVIEW</p> <p>Zeigt den nächsten fälligen Termin an. <input type="radio"/> <input type="button" value="←"/> TIME <input type="button" value="←"/> REVIEW</p> <p>In anderen Menüs: Listet die Namen und Arten der Operationen auf. <input type="radio"/> <input type="button" value="←"/> REVIEW</p>	<p>396</p> <p>316</p> <p>326</p> <p>275</p> <p>312</p> <p>285</p> <p>205</p> <p>472</p> <p>120</p>

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
RL	Rotiert um ein Bit nach links. C [MTH] BASE p.2 RL	227
RLB	Rotiert um ein Byte nach links. C [MTH] BASE p.2 RLB	227
RND	Rundet die Nachkommastellen einer Zahl oder eines Namens. F [MTH] PARTS p.4 RND	160
RNRM	Ermittelt die Zeilennorm einer Matrix (größte Zeilensumme). C [MTH] MATR p.2 RNRM	387
ROLL	Verschiebt das Objekt in Ebene ($n + 1$) nach Ebene 1. C [PRG] STK ROLL	84
ROLL	Rollt das Objekt aus der aktuellen Ebene nach Ebene 1. O ... [↑STK] ROLL	76
ROLLD	Verschiebt das Objekt in Ebene 2 nach Ebene n . C [PRG] STK ROLLD	84
ROLLD	Verschiebt das Objekt in Ebene 1 zur aktuellen Ebene. O ... [↑STK] ROLLD	76
ROOT	Löst eine Gleichung nach einer unbekannten Variablen auf. C [←][SOLVE] ROOT	275

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ROOT	Bewegt den Grafikcursor zum Schnittpunkt einer Funktion mit der x-Achse, zeigt den Wert der Nullstelle an und überträgt den Wert in den Stack. O  GRAPH FCN ROOT	331
ROT	Verschiebt das Objekt in Ebene 3 nach Ebene 1. C  STK ROT	84
+ROW	Fügt an der aktuellen Zeile eine Zeile mit Nullen ein. O  MATRIX p.2 +ROW	377
-ROW	Löscht die aktuelle Zeile. O  MATRIX p.2 -ROW	377
RPT	Wählt das Menü TIME ALRM RPT (Terminwiederholung). O  TIME ALRM RPT	
RR	Rotiert um ein Bit nach rechts. C  BRSE p.2 RR	227
RRB	Rotiert um ein Byte nach rechts. C  BRSE p.2 RRB	227
RSD	Berechnet einen Korrekturwert zur Lösung eines Gleichungssystems. C  MRTR RSD	389
RULES	Aktiviert das Transformationsmenü RULES für ein angegebenes Objekt. O  EQUATION  RULES	428

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
R→B	Umwandlung einer reellen Zahl in eine Binärzahl. C [MTH] BASE p.2 R→B	226
R→C	Umwandlung einer reellen Zahl in eine komplexe Zahl. C [PRG] OBJ pg.2 R→C	102
R→D	Umwandlung von Radiant in Grad. F [MTH] VECTR p.2 R→D	152
R↵Z	Wählt den Modus Polar-/Zylinderkoordinaten. [MTH] VECTR R↵Z O [↵] [MODES] p.3 R↵Z	183
R↵↵	Wählt den Modus Polar-/Kugelkoordinaten. [MTH] VECTR R↵↵ O [↵] [MODES] p.3 R↵↵	183
s	Sekunde, Zeit (1 s). U [↵] [UNITS] TIME S	
S	Siemens, elektrischer Leitwert (1 A ² ·s ³ /kg·m ²). U [↵] [UNITS] p.2 ELEC p.2 S	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
SAME	Testet zwei Objekte auf Gleichheit. C [PRG] TEST SAME	529
sb	stilb, Leuchtdichte (10000 cd/m ²) U [↩] [UNITS] p.3 LIGHT SB	
SBRK	Sendet ein BREAK über den seriellen Anschluß. C [↩] [I/O] p.3 SBRK	687
SCALE	Legt den Maßstab für die PLOT-Achsen fest. ... PLOTR p.2 SCALE C [→] [PLOT] p.2 SCALE [→] [SCALE] Legt den gültigen Maßstab im Stack ab. ... PLOTR p.2 [→] SCALE O [→] [PLOT] p.2 [→] SCALE	317 316
SCATRLOT	Erzeugt ein Streudiagramm der statistischen Daten in Σ DAT. C [↩] [STAT] p.3 SCATR	407
SCATTER	Wählt den Plottyp SCATTER. C ... PTYPE p.2 SCATT	352

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
SCI	Stellt für die Anzeige den SCI-Modus ein. C   SCI	62
SCONJ	Konjugiert den Inhalt einer Variablen. C   p.2 SCON	124
SDEV	Berechnet die Standardabweichung . C   p.2 SDEV	402
	Stellt das Wiederholintervall für Termine auf n Sekunden ein. O   ALRM RPT 	478
	Stellt die Uhr um 1 Sekunde vor. O   ADJUST 	476
	Stellt die Uhr um 1 Sekunde zurück. O   ADJUST 	476
SEND	Sendet den Inhalt einer Variablen an ein anderes Gerät. C   SEND	666
SERVER	Bringt den HP 48 in den Kermit-Servermodus.   SERV C  	666

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
SET	Wählt das Menü TIME SET. O \leftarrow [TIME] SET	
SET	Setzt einen Termin fest. O \leftarrow [TIME] ALRM SET	478
SETUP	Wählt das Menü I/O SETUP (Einstellung der E/A-Parameter). O \leftarrow [I/O] SETUP	
SF	Setzt ein angegebenes Flag. [PRG] TEST p.3 SF C \rightarrow [MODES] p.2 SF	556
SHOW	Formt einen Ausdruck so um, daß auch verdeckte Variablen, die nur implizit in anderen Variablen vorkommen, sichtbar werden. C \leftarrow [ALGEBRA] SHOW	423
SIGN	Liefert das Vorzeichen einer Zahl. F [MTH] PARTS SIGN	160
SIN	Sinus. A [SIN]	150
SINH	Sinus hyperbolicus. A [MTH] HYP SINH	147

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
SINV	Ersetzt den Inhalt einer Variablen durch ihren Kehrwert. C  MEMORY p.2 SINV	124
SIZE	Bestimmt die Dimensionen einer Liste eines Feldes, einer Zeichenkette, eines algebraischen Objektes oder eines Grafikobjektes.  OBJ p.3 SIZE C  DSPL p.2 SIZE	100 367
←SKIP	Verschiebt den Cursor nach links zum Anfang eines Wortes.  EDIT ←SKIP O ... EDIT ←SKIP	73
SKIP→	Verschiebt den Cursor nach rechts zum Anfang des nächsten Wortes.  EDIT SKIP→ O ... EDIT SKIP→	73
SL	Schiebt um ein Bit nach links. C MTH BASE p.3 SL	227

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
SLB	Schiebt um ein Byte nach links. C [MTH] BASE p.3 [SLB]	227
SLOPE	Berechnet die Steigung einer Funktion an der Cursorposition, zeigt sie an und überträgt den Wert in den Stack. O ... [FCN] [SLOPE]	331
slug	slug, Masse (14,5939029372 kg). U [←] [UNITS] MASS [SLUG]	
SNEG	Negiert den Inhalt einer Variablen. C [→] [MEMORY] p.2 [SNEG]	124
[SOLVE]	Wählt das Menü SOLVE. O [←] [SOLVE]	
SOLVR	Wählt das Menü SOLVR. [←] [SOLVE] SOLVR [←] [SOLVE] CAT SOLVR [→] [SOLVE] [←] [PLOT] CAT SOLVR O [→] [ALGEBRA] SOLVR	
[SPC]	Gibt eine Leerstelle in die Befehlszeile ein. O [SPC]	
SPEED	Wählt das Menü UNITS SPEED. O [←] [UNITS] SPEED	

Name, Taste oder Menüfeld	Beschreibung Art, Taste	Seite
SQ	Quadriert das Objekt in Ebene 1. A  	144
SR	Schiebt um ein Bit nach rechts. C   p.3 	227
sr	Steradian, Raumwinkel ($7,95774715459 \times 10^{-2}$). U   p.3  	
SRB	Schiebt um ein Byte nach rechts. C   p.3 	227
SRECV	Liest eine angegebene Anzahl von Zeichen vom E/A-Anschluß. C   p.3 	686
	Führt den nächsten Schritt in einem unterbrochenen Programm aus. Führt ein aufgerufenes Unterprogramm jedoch vollständig aus. O   	519
 ↓	Führt den nächsten Befehlsschritt in einem unterbrochenen Programm aus. Führt auch Unterprogramme schrittweise aus. O    ↓	519
st	Stere, Volumen (1 m^3). U    	
St	stokes, kinematische Viskosität ($0,0001 \text{ m}^2/\text{s}$) U   p.3  	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
START	Beginnt eine bestimmte Schleife. C [PRG] [BRCH] [START]	541
[←] START	Gibt START NEXT ein. O [PRG] [BRCH] [←] START	541
[→] START	Erzeugt START STEP als Schreibhilfe. O [PRG] [BRCH] [→] START	543
[←] [STAT]	Wählt das Menü STAT (Statistik). O [←] [STAT]	
[→] [STAT]	Wählt Seite 2 des Menüs STAT. O [→] [STAT]	
STD	Wählt den Standardanzeigemodus. C [←] [MODES] [STD]	62
STEP	Beendet eine bestimmte Schleife. C [PRG] [BRCH] p.2 [STEP]	543
		547

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
STEQ	Speichert eine Gleichung, die in Ebene 1 steht, in EQ. ⬅ PLOT STEQ ... PLOTR ⬅ DRAW ➡ PLOT ⬅ DRAW C ⬅ SOLVE STEQ	276
STIME	Legt fest, nach welcher Wartezeit eine serielle Übertragung abgebrochen wird. C ⬅ I/O p.3 STIME	686
STK	Wählt das Menü PRG STK (Programmstack). O PRG STK	
STK	Schaltet die Sicherung des letzten Stackinhalts ein und aus. O ⬅ MODES p.2 STK	237
↑STK	Wählt den Interaktiven Stack. ... EDIT ↑STK ⬅ EDIT ↑STK ➡ VISIT ↑STK O ➡ MATRIX p.2 ↑STK	75 377

Name, Taste oder Menüfeld	Description Type, Keys	Page
→STK	<p>Kopiert die gewählte Gleichung nach Ebene 1.</p> <p>... CAT p.2 →STK</p> <p>O [↩] ALGEBRA p.2 →STK</p> <p>Kopiert die gewählte Matrix nach Ebene 1.</p> <p>O [↩] STAT CAT →STK</p> <p>Kopiert den gewählten Termin nach Ebene 1.</p> <p>[↩] TIME CAT →STK</p> <p>O [↩] TIME →STK</p> <p>Kopiert das gewählte Matrixelement nach Ebene 1.</p> <p>O [↩] MATRIX p.2 →STK</p>	<p>279</p> <p>399</p> <p>484</p> <p>377</p>
STO	<p>Speichert ein Objekt in einer Variablen.</p> <p>C [STO]</p>	114
[STO]	<p>Speichert ein Objekt in einer Variablen und sichert den vorherigen Inhalt der Variablen, so daß er durch LASTARG wiederhergestellt werden kann.</p> <p>O [STO]</p>	114
[STO]	<p>Legt eine EquationWriter-Gleichung oder PICT im Stack ab.</p> <p>O [STO]</p>	<p>246</p> <p>326</p>
STOALARM	<p>Speichert einen Termin, der in Ebene 1 steht, in der Terminliste.</p> <p>C [↩] TIME ALRM p.2 STOAL</p>	487
STOF	<p>Legt den Status der System- und der Benutzerflags fest.</p> <p>C [↩] MODES p.2 STOF</p>	558

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
STOKEYS	Belegt mehrere Tasten mit benutzerspezifischen Definitionen. C  MODES  STOK	233
STO+	Addiert eine Zahl oder ein Feld zum Inhalt einer Variablen. C  MEMORY  STO+	123
STO-	Subtrahiert eine Zahl oder ein Feld vom Inhalt einer Variablen. C  MEMORY  STO-	124
STO*	Multipliziert den Inhalt einer Variablen mit einer Zahl. C  MEMORY  STO*	124
STO/	Dividiert den Inhalt einer Variablen durch eine Zahl. C  MEMORY  STO/	124
STO Σ	Speichert die aktuelle Statistikmatrix in Σ DATA. C  STAT  STOΣ	396
STR→	Konvertiert eine Zeichenkette in Komponentenobjekte. C Bitte eintippen.	
→STR	Konvertiert ein Objekt in eine Zeichenkette C PRG  OBJ  →STR	102
STWS	Legt die Wortlänge von Binärzahlen fest. C MTH  BASE  STWS	222

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
SUB	Erstellt einen Auszug aus einer Liste, einer Zeichenkette oder einem Grafikobjekt. PRG OBJ p.3 SUB C PRG DSPL p.3 SUB	103 368
SUB	Legt einen angegebenen Teil von <i>PICT</i> im Stack ab. ... DRAW p.3 SUB ... AUTO p.3 SUB O ← GRAPH p.3 SUB	366
SUB	Legt einen angegebenen Teilausdruck im Stack ab. O ← EQUATION ← SUB	428
Sv	Sievert, Dosisäquivalent ($0,01 \text{ m}^2/\text{s}^2$) U ← UNITS p.3 RAD SV	
SWAP	Tauscht Objekte in den Ebenen 1 und 2 aus. C ← SWAP	67
SYM	Schaltet zwischen dem Modus für symbolische und dem Modus für numerische Ergebnisse um. O ← MODES SYM	155
SYSEVAL	Berechnet ein Systemobjekt. <i>Nur verwenden, wenn durch ein HP-Anwendungsprogramm angegeben.</i> C Bitte eintippen.	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
t	Metrische Tonne, Masse (1000 kg). U [UNITS] MASS p.2 T	
T	Tesla, Magnetischer Fluß (1 kg/As ²). U [UNITS] p.2 ELEC p.2 T	
T	Ausdruck nach links verschieben. O [EQUATION] RULES T Führt T aus, bis keine Änderung mehr im Teilausdruck auftritt. O [EQUATION] RULES T	432 440
T T	Ausdruck nach rechts verschieben. O [EQUATION] RULES T Führt T aus, bis keine Änderung mehr im Teilausdruck auftritt. O [EQUATION] RULES T	432 440
%T	Liefert den prozentualen Anteil des Wertes in Ebene 1 am Gesamtwert in Ebene 2. F [MTH] PARTS p.2 T	148
→TAG	Verbindet die Objekte in den Ebenen 1 und 2 zu einem markierten Objekt. C [PRG] OBJ →TAG	101
TAN	Tangens. A [TAN]	150

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
TANH	Tangens hyperbolicus. A [MTH] HYP TANH	147
TAYLR	Berechnet eine Taylorreihe. C [←] [ALGEBRA] TAYLR	458
tbsp	tablespoon, Volumen (1,47867647813 × 10 ⁻⁵ m ³). U [←] [UNITS] VOL p.3 TBSP	
TEMP	Wählt das Menü UNITS TEMP (Temperatur). O [←] [UNITS] p.2 TEMP	
TEST	Wählt das Menü PRG TEST (Programmtest). O [PRG] TEST	
TEXT	Stellt die Stackanzeige dar . C [PRG] DSPL p.4 TEXT	369
THEN	Beginnt die THEN-Anweisung. C [PRG] BRCH p.2 THEN	532
therm	Wärmeeinheit EEC, Energie (105506000 kg·m ² /s ²) U [←] [UNITS] p.2 ENRG p.2 THER	
TICKS	Liefert die Systemzeit als Binärzahl, d.h. als Anzahl von Uhrzyklen. C [←] [TIME] p.2 TICKS	490

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
TIME	Liefert die aktuelle Zeit. C <input type="button" value="←"/> TIME p.2 TIME	490
<input type="button" value="←"/> TIME <input type="button" value="→"/> TIME	Wählt das Menü TIME. O <input type="button" value="←"/> TIME Wählt den Terminkatalog. O <input type="button" value="→"/> TIME	482
TIME	Wählt das Menü UNITS TIME. O <input type="button" value="←"/> UNITS TIME	
→TIME	Stellt die Systemzeit ein. C <input type="button" value="←"/> TIME SET →TIM	475
>TIME	Stellt die Terminzeit ein. O <input type="button" value="←"/> TIME ALRM >TIME	478
TLINE	Wie LINE, die Gerade wird jedoch vor beliebigem Hintergrund dargestellt, d.h. die Bildpunkte werden umgeschaltet und nicht einfach eingeschaltet. C <input type="button" value="PRG"/> DSPL TLINE	364

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
TLINE	<p>Wie LINE, die Gerade wird jedoch vor beliebigem Hintergrund dargestellt, d.h. die Bildpunkte werden umgeschaltet und nicht einfach eingeschaltet.</p> <p>... DRAW p.2 TLINE ... AUTO p.2 TLINE O  GRAPH p.2 TLINE</p>	362
TMENU	<p>Erzeugt ein vorübergehend gültiges Menü, dessen Inhalt durch eine Liste angegeben wird. Der Inhalt von CST wird nicht verändert.</p> <p>C  MODES p.2 TMEN</p>	583
ton	<p>short ton, Masse (907,18474 kg).</p> <p>U  UNITS MASS p.2 TON</p>	
tonUK	<p>long (UK) ton, Masse (1016,0469088 kg).</p> <p>U  UNITS MASS p.2 TONU</p>	
torr	<p>Torr (mmHg), Druck (133,322368421 kg/ms²).</p> <p>U  UNITS p.2 PRESS TORR</p>	
TOT	<p>Bildet die Summe jeder Spalte der Matrix in ΣDAT.</p> <p>C  STAT p.2 TOT</p>	402
TRANSIO	<p>Wählt eine von drei Einstellungen zur Zeichenumsetzung bei der Ein-/Ausgabe.</p> <p>C  I/O SETUP TRAN</p>	670

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
TRG*	Formt trigonometrische Funktionen und Hyperbelfunktionen von Summen und Differenzen um. O  EQUATION  RULES TRG*	439
TRN	Transponiert eine Matrix. C   	387
TRNC	Rundet eine Zahl in Ebene 2 auf die in Ebene 1 angegebene Stellenzahl ab. F  PARTS p.4 	160
TRUTH	Wählt den Plottyp TRUTH. C ...  TRUTH	351
tsp	teaspoon, Volumen ($4,92892159375 \times 10^{-6} \text{ m}^3$). U    p.3 	
TSTR	Konvertiert Datum und Zeit von der Zahlendarstellung in eine Zeichenkette. C   p.2 	489
TVARS	Listet Variable auf, die Objekte eines angegebenen Typs enthalten. C   p.2 	103
TYPE	Liefert die Typnummer eines angegebenen Objekts.   p.2  C   	102
u	Atom-Masseneinheit ($1,66057 \times 10^{-27} \text{ kg}$). U    p.3 	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
UBASE	Konvertiert die Einheiten eines Objektes in SI-Einheiten. F  UNITS UBASE	210
UFACT	Wandelt zusammengesetzte Einheiten so um, daß eine bestimmte Einheit als Faktor darin enthalten ist. C  UNITS UFACT	213
→UNIT	Verbindet Objekte in den Ebenen 1 und 2 zu einem Objekt mit Einheiten.  OBJ p.2 →UNIT C  UNITS →UNIT	101
 UNITS  UNITS	Wählt das Menü UNITS-Katalogmenü. O  UNITS Wählt das Menü UNITS-Befehlsmenü. O  UNITS	
UNTIL	Beginnt die UNTIL Anweisung. C  BRCH p.2 UNTIL	549
UPDIR	Macht das übergeordnete Verzeichnis zum aktuellen Verzeichnis. C  UP	131
	Schaltet den Benutzermodus ein und aus. O  USR	232
UTPC	Liefert die Wahrscheinlichkeit, daß die chi-Quadrat verteilte Zufallsgröße größer ist als x. C  PROB p.2 UTPC	413

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
UTPF	Liefert die Wahrscheinlichkeit zurück, daß die Snedecor F-verteilte Zufallsgröße größer ist als x. C [MTH] PROB p.2 UTPF	415
UTPN	Liefert die Wahrscheinlichkeit, daß eine normalverteilte Zufallsgröße größer ist als x. C [MTH] PROB p.2 UTPN	415
UTPT	Liefert die Wahrscheinlichkeit, daß eine Standard-t-verteilte Zufallsgröße größer ist als x. C [MTH] PROB p.2 UTPT	415
UVAL	Sendet den Zahlenwert eines angegebenen Objektes mit Einheiten. F [→] [UNITS] UVAL	221
V	Volt, elektrische Potentialdifferenz ($1 \text{ kg} \cdot \text{m}^2 / \text{A} \cdot \text{s}^3$). U [←] [UNITS] p.2 ELEC V	
[VAR]	Wählt das Menü VAR (Variable). O [VAR]	120
1-VAR	Macht den gewählten Eintrag zur aktuellen statistischen Matrix und stellt die zweite Seite des Menüs STAT dar. O [←] [STAT] CAT 1-VAR	399
2-VAR	Macht den gewählten Eintrag zur aktuellen statistischen Matrix und stellt die vierte Seite des Menüs STAT dar. O [←] [STAT] CAT 2-VAR	399
VARS	Liefert eine Liste der Variablen im aktuellen Verzeichnis. C [←] [MEMORY] VARS	121

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
VISIT	Ist das Argument ein Name, wird der Inhalt der zugehörigen Variablen zur Bearbeitung in die Befehlszeile kopiert. Ist das Argument eine Stackebenennummer, wird das Objekt in der bezeichneten Ebene zur Bearbeitung in die Befehlszeile kopiert. O  VISIT	71
VOL	Wählt das Menü UNITS VOL (Volumen). O  UNITS VOL	
VTYPE	Liefert die Typnummer eines Objektes, das unter einem lokalen oder einem globalen Namen gespeichert ist. C  PRG OBJ p.2 VTYPE	104
→V2	Faßt zwei reelle Zahlen zu einem zweidimensionalen Vektor oder zu einer komplexen Zahl zusammen. C  MTH VECTR p.2 →V2	179
→V3	Faßt drei reelle Zahlen zu einem dreidimensionalen Vektor zusammen. C  MTH VECTR p.2 →V3	196
V→	Zerlegt einen Vektor aus 2 oder 3 Elementen entsprechend dem aktuellen Winkelmodus. C  MTH VECTR p.2 V→	179
W	Watt, Leistung ($1 \text{ kg} \cdot \text{m}^2 / \text{s}^3$)  UNITS p.2 POWR W U  UNITS p.2 ELEC W	
*W	Legt den horizontalen Plotmaßstab fest. C ... PLOTR p.3 *W	343

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
WAIT	Unterbricht Programmausführung für eine angegebene Anzahl von Sekunden oder bis eine Taste gedrückt wird. C [PRG] CTRL p.2 WAIT	578
Wb	Weber, Magnetischer Fluß (1 kg·m ² /A·s ²). U [←] [UNITS] p.2 ELEC p.2 WB	
WEEK	Stellt das Wiederholintervall für Terminerinnerungen auf <i>n</i> Wochen ein. O [←] [TIME] ALRM RPT WEEK	478
WHILE [←]WHILE	Beginnt eine unbestimmte Schleife. C [PRG] BRCH WHILE Erzeugt WHILE REPEAT END als Schreibhilfe. O [PRG] BRCH [←]WHILE	551 551
WID→	Erhöht die Spaltenbreite und verringert die Anzahl der angezeigten Spalten. O [→] [MATRIX] WID→	378
←WID	Verringert die Spaltenbreite und erhöht die Anzahl der angezeigten Spalten. O [→] [MATRIX] ←WID	378
X	Übernimmt den angegebenen Wert als Zoom-Faktor für die x-Achse. O ... ZOOM X	328
ΣX	Liefert die Summe der Daten in einer unabhängigen Spalte in ΣDAT zurück. C [←] [STAT] p.5 ΣX	414

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ΣX^2	Liefert die Summe der quadrierten Daten in einer unabhängigen Spalte in ΣDAT zurück. C \leftarrow [STAT] p.5 ΣX^2	414
$\Sigma AUTO$	Übernimmt den angegebenen Wert als Zoom-Faktor für die x-Achse und wählt den Maßstab der y-Achse automatisch. O ... ZOOM $\Sigma AUTO$	328
ΣCOL \rightarrow ΣCOL	Legt die Spalte der unabhängigen Variablen in der Matrix in ΣDAT fest. C \leftarrow [STAT] p.3 ΣCOL Holt die Spaltennummer der unabhängigen Variablen in den Stack. O \leftarrow [STAT] p.3 \rightarrow ΣCOL	406 406
XMIT	Seriell Senden von Zeichenketten über den seriellen Anschluß ohne Kermit-Protokoll. C \leftarrow [I/O] p.3 XMIT	685
XOR	Logisches oder binäres Exklusiv-ODER. [MTH] BASE p.4 XOR F [PRG] TEST XOR	227 531
XPON	Liefert den Exponenten einer Zahl zurück. F [MTH] PARTS p.3 XPON	160
XRNG	Bestimmt den Anzeigebereich der x-Achse. ... PLOTR XRNG C \rightarrow [PLOT] XRNG	317

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
XRNG	Ruft den Anzeigebereich der x-Achse in den Stack zurück. ... PLOTR XRNG O [PLOT] XRNG	315
XROOT	Liefert die xte (x in Ebene 1) Wurzel einer Zahl, die in Ebene 2 steht. A $\sqrt[x]{y}$	144
$\times Y$	Übernimmt den angegebenen Wert als Zoom-Faktor für die x- und die y-Achse. O ... ZOOM $\times Y$	328
$\times Y Z$	Wählt den Rechteckkoordinaten-Modus. [MTH] VECTR $\times Y Z$ O [MODES] p.3 $\times Y Z$	184
$\Sigma X * Y$	Liefert die Summe der Produkte von Daten in einer unabhängigen und einer abhängigen Spalte in ΣDAT . C [STAT] p.5 $\Sigma X * Y$	414
$\times Y$	Übernimmt den angegebenen Wert als Zoom-Faktor für die y-Achse. O ... ZOOM $\times Y$	328
ΣY	Liefert die Summe der Daten in einer unabhängigen Spalte in ΣDAT . C [STAT] p.5 ΣY	414
ΣY^2	Liefert die Summe der quadrierten Daten in einer abhängigen Spalte in ΣDAT zurück. C [STAT] p.5 ΣY^2	414

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
YCOL	Legt die angegebene Spalte der Matrix in ΣDAT als Spalte der abhängigen Variablen bei Zwei-Variablen-Statistik fest.	406
 YCOL	C  [STAT] p.3 YCOL Holt die Spaltennummer für die abhängigen Variablen in den Stack. O  [STAT] p.3  YCOL	406
yd	yard, Länge (0,9144 m). U  [UNITS] LENG YD	
yd ²	square yard, Fläche (0,83612736 m ²). U  [UNITS] ARER YD ²	
yd ³	cubic yard, Volumen (0,764554857984 m ³). U  [UNITS] VOL YD ³	
yr	Jahr, Zeit (31556925,9747 s). U  [UNITS] TIME YR	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
<p>YRNG</p> <p>☞ YRNG</p>	<p>Bestimmt den Anzeigebereich der y-Achse.</p> <p>... PLOTR YRNG</p> <p>C ☞ PLOT YRNG</p> <p>Ruft den Anzeigebereich der y-Achse in den Stack zurück.</p> <p>... PLOTR ☞ YRNG</p> <p>O ☞ PLOT ☞ YRNG</p>	<p>315</p> <p>315</p>
<p>Z-BOX</p> <p>☞ Z-BOX</p>	<p>Plottet die Grafik neu, wobei das Rechteck, dessen gegenüberliegende Ecken durch die Marke und den Cursor definiert sind, das Display ausfüllt.</p> <p>... DRAW Z-BOX</p> <p>... AUTO Z-BOX</p> <p>O ☞ GRAPH Z-BOX</p> <p>Plottet die Grafik neu, wobei der horizontale Abstand zwischen Marke und Cursor das Display in x-Richtung ganz ausfüllt. Die y-Achse wird automatisch skaliert.</p> <p>... DRAW ☞ Z-BOX</p> <p>... AUTO ☞ Z-BOX</p> <p>O ☞ GRAPH ☞ Z-BOX</p>	<p>329</p> <p>329</p>

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
ZOOM	Wählt das Menü GRAPHICS ZOOM. ... DRAW ZOOM ... AUTO ZOOM O  GRAPH ZOOM	327
+	Addiert zwei Objekte. A 	95
	Steht der Cursor auf einer Zahl, wird das Vorzeichen der Mantisse oder des Exponenten dieser Zahl geändert. Wirkt ansonsten als NEG-Taste. C 	49
	Schaltet die Darstellung des Cursors um zwischen dunkel und invertiert, d.h. dunkel auf hellem Hintergrund und hell auf dunklem Hintergrund. ... DRAW p.3  ... AUTO p.3  O  GRAPH p.3 	324
	Addiert und Subtrahiert 1. O  EQUATION  RULES 	435
-	Subtrahiert zwei Objekte. A 	144

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
- ()	Negiert zweimal und teilt auf. O  EQUATION  RULES - ()	441
*	Multipliziert zwei Objekte. A 	144
* 1	Multipliziert mit 1. O  EQUATION  RULES * 1	435
/	Dividiert zwei Objekte. A 	144
/ 1	Dividiert durch 1. O  EQUATION  RULES / 1	431
^	Erhebt eine Zahl zur angegebenen Potenz. A 	144
^ 1	Erhebt eine Zahl zur ersten Potenz. O  EQUATION  RULES ^ 1	435
<	Vergleichsfunktion "Kleiner als".  TEST p.2 < F   2	529

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
≤	Vergleichsfunktion "Kleiner als oder gleich". PRG TEST p.2 ≤ F α ↵ 3	529
>	Vergleichsfunktion "Größer als". PRG TEST p.2 > F α → 2	529
≥	Vergleichsfunktion "Größer als oder gleich". PRG TEST p.2 ≥ F α → 3	529
=	Gleichheitszeichen. A ↵ =	138
==	Vergleichsfunktion "Gleich". PRG TEST p.2 == F α ↵ 1	530
≠	Vergleichsfunktion "Ungleich". PRG TEST p.2 ≠ F α ↵ 0	530
α	Schaltet den alphabetischen Eingabemodus ein und aus. O α	56
{ }	Schaltet implizite Klammern ein und aus. O ↵ EQUATION ↵ { }	255

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
□ □	Legt Gleichung als Zeichenkette im Stack ab. O  EQUATION  □ □	247
,	Zerlegt komplexe Zahlen oder Funktionsargumente in ihre Elemente. C  □	170
°	Grad, Winkel in der Ebene ($2,7777777778 \times 10^{-3}$). U  UNITS p.3 ANGL  □	
!	Fakultät.  PROB  ! F   DEL	158
∫	Integral. A  ∫	462
∂	Ableitung. A  ∂	453
Ω	Ohm, elektrischer Widerstand ($1 \text{ kg}\cdot\text{m}^2/\text{A}^2\cdot\text{s}^3$). U  UNITS p.2 ELEC  Ω	
%	Liefert einen Prozentwert, wenn der Prozentsatz in Ebene 2 und der Gesamtwert in Ebene 1 stehen. A  PARTS p.2  %	148
Σ	Summenbildung. F  Σ	458

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
$\Sigma+$	Addiert einen Datenpunkt zur Matrix in ΣDAT . C \leftarrow [STAT] $\Sigma+$	396
$\Sigma-$	Subtrahiert einen Datenpunkt von einer Matrix in ΣDAT . C \leftarrow [STAT] \leftarrow $\Sigma+$	396
$\sqrt{\quad}$	Liefert die Quadratwurzel eines Objektes in Ebene 1. A \sqrt{x}	144
	Fügt einem berechneten Ausdruck einen lokalen Namen oder eine Integrationsvariable zusammen mit deren Wert hinzu. F \leftarrow [ALGEBRA] p.2	450
$1/()$	Invertiert zweimal und teilt auf. O \leftarrow [EQUATION] \leftarrow RULES $1/()$	441
12/24	Schaltet zwischen dem 12-Stunden- und dem 24-Stunden-Anzeigeformat um. O \leftarrow [TIME] SET 12/24	476

Name, Taste oder Menüfeld	Beschreibung Art, Taste	Seite
	Klammert benachbarte Größen ein. O 	437
 	Erweitert Teilausdruck nach links. O  Führt  aus, bis keine Änderung mehr im Klammerausdruck auftritt. O 	438 444
	Löst Klammerausdruck auf. O 	439
 	Erweitert Klammerausdruck nach rechts. O  Führt  aus, bis keine Änderung mehr im Klammerausdruck auftritt. O 	438 444
	Vertauscht Argumente. O 	438
	Neue Zeile. C 	
	Erzeugt eine lokale Variable. C 	510

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
	Umschalttaste links. O 	55
	Umschalttaste rechts. O 	55
	In einer Befehlszeile wird das Zeichen links vom Cursor gelöscht. O  Löscht den Inhalt der aktuellen Stackebene. O ...  STK 	49 77
	Bewegt den Cursor in mehrzeiligen Befehlszeilen um eine Zeile nach oben. Bewegt den Zeiger im Interaktiven Stack um eine Ebene nach oben. Bewegt den Cursor in der Grafikumgebung um einen Bildpunkt nach oben. Bewegt das Fenster im Rollmodus um einen Bildpunkt nach oben. Bewegt den Zellencursor im MatrixWriter um eine Zeile nach oben. Beginnt den Zähler im EquationWriter. Bewegt den Cursor in der Auswahlumgebung um ein Objekt nach oben. Bewegt den Zeiger in Katalogen um einen Eintrag nach oben. O 	80 77 326 246 376 246 432

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
	Bewegt den Cursor in mehrzeiligen Befehlszeilen zur obersten Zeile.	80
	Bewegt den Zeiger im Interaktiven Stack zur höchsten Stackebene.	77
	Bewegt den Cursor in der Grafikumgebung zur Oberkante von <i>PICT</i> .	326
	Bewegt den Zellencursor im MatrixWriter zum obersten Element der aktuellen Spalte.	376
	Bewegt den Cursor in der Auswahlumgebung zum obersten Objekt.	432
	Bewegt den Cursor in Katalogen an die erste Stelle einer Liste. ○ 	
	Bewegt den Zeiger in Katalogen um eine Seite nach oben.	77
	Bewegt den Zeiger im Interaktiven Stack um 4 Ebenen nach oben. ○ 	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
▼	<p>Bewegt den Cursor in mehrzeiligen Befehlszeilen um eine Zeile nach unten. 80</p> <p>Bewegt den Zeiger im Interaktiven Stack um eine Ebene nach unten. 77</p> <p>Bewegt den Cursor in der Grafikumgebung um einen Bildpunkt nach unten. 325</p> <p>Bewegt ein Fenster im Rollmodus um einen Bildpunkt nach unten. 246</p> <p>Bewegt den Zellencursor im MatrixWriter um eine Zeile nach unten. 376</p> <p>Beendet einen Teilausdruck im EquationWriter. 246</p> <p>Bewegt den Cursor in der Auswahlumgebung um ein Objekt nach unten. 432</p> <p>Bewegt den Zeiger in Katalogen um einen Eintrag nach unten.</p> <p>○ ▼</p>	
→ ▼	<p>Bewegt den Cursor in mehrzeiligen Befehlszeilen zur untersten Zeile. 80</p> <p>Bewegt den Zeiger im Interaktiven Stack zur Ebene 1. 77</p> <p>Bewegt den Cursor in der Grafikumgebung zur oberen Kante von <i>PICT</i>. 325</p> <p>Bewegt den Zellencursor im MatrixWriter zum letzten Element der aktuellen Spalte. 376</p> <p>Beendet alle Teilausdrücke im EquationWriter 246</p> <p>Bewegt den Cursor in der Auswahlumgebung zum untersten Objekt. 432</p> <p>Bewegt den Zeiger in Katalogen zum Ende einer Liste.</p> <p>○ → ▼</p>	

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
	<p>Bewegt den Zeiger in Katalogen um eine Seite nach unten.</p> <p>Bewegt den Zeiger im Interaktiven Stack um 4 Ebenen nach unten.</p> <p>○ </p>	77
	<p>Bewegt den Zeiger in der Befehlszeile um ein Zeichen nach links.</p> <p>Bewegt den Cursor in der Grafikumgebung um einen Bildpunkt nach links.</p> <p>Bewegt das Fenster im Rollmodus um einen Bildpunkt nach links.</p> <p>Bewegt den Zellencursor im MatrixWriter um eine Spalte nach links.</p> <p>Aktiviert die Auswahlumgebung im EquationWriter.</p> <p>Bewegt den Cursor in der Auswahlumgebung um ein Objekt nach links.</p> <p>○ </p>	80 325 246 376 428 432
 	<p>Ruft den Rollmodus im EquationWriter und in der Grafikumgebung auf.</p> <p>○  ()</p>	246 326

Name, Taste oder Menüfeld	Beschreibung Art, Tasten	Seite
	<p>Bewegt den Cursor in der Befehlszeile an den Anfang der aktuellen Zeile. 80</p> <p>Bewegt den Cursor in der Grafikumgebung an die linke Kante von <i>PICT</i>. 325</p> <p>Bewegt den Zellencursor im MatrixWriter zum ersten Element der aktuellen Zeile. 376</p> <p>Bewegt den Cursor in der Auswahlumgebung zum ganz linken Objekt. 432</p> <p>○ </p>	
	<p>Bewegt den Cursor in der Befehlszeile um ein Zeichen nach rechts. 80</p> <p>Bewegt den Cursor in der Grafikumgebung um einen Bildpunkt nach rechts. 325</p> <p>Bewegt ein Fenster im Rollmodus um einen Bildpunkt nach rechts. 246</p> <p>Bewegt den Zellencursor im MatrixWriter um eine Spalte nach rechts. 376</p> <p>Beendet einen Teilausdruck im EquationWriter. 246</p> <p>Bewegt den Cursor in der Auswahlumgebung um ein Objekt nach rechts. 432</p> <p>○ </p>	

Name, Taste oder Menüfeld	Beschreibung Art, Taste	Seite
 	<p>Bewegt den Cursor in einer Befehlszeile an das Ende der aktuellen Zeile. 80</p> <p>Bewegt den Cursor in der Grafikumgebung zur rechten Kante von <i>PICT</i>. 325</p> <p>Bewegt den Zellencursor im MatrixWriter zum letzten Element der aktuellen Zeile. 376</p> <p>Beendet alle Teilausdrücke im EquationWriter. 246</p> <p>Bewegt den Cursor in der Auswahlumgebung zum ganz rechten Objekt. 432</p> <p>○  </p>	

Index

A

- Abbrechen mit der ATTN-Taste
 - Befehlszeile, 58
 - Programme, 58
 - Umgebungen, 58
- abgelaufene Zeit, 494
- abgezählte Zeichenketten, 91
- abhängige Variable
 - Diagramme von Funktionen, 323
 - Kegelschnitte plotten, 355
 - Plotbereich, 345
 - Plottyp TRUTH, 359
 - Statistik, 406
- Ablauf, Programme, 508
- Ableitungen
 - benutzerdefinierte Ableitungen, 456
 - Eintippen von Ableitungen in die Befehlszeile, 454
 - EquationWriter, 252
 - plotten, 332
- Abschaltzeit, 25
- Achsen
 - Beschriftungen, 346
 - Koordinatenursprung festlegen, 346
- addieren
 - einen Wert aus dem Stack zu einer Variablen, 123
 - EquationWriter, 248
 - Zahlen, 144
- ADJST-Menü für Uhrzeit, 478
- aktueller Pfad, 127
- aktuelles Verzeichnis, 127
- akustisches Signal, aus einem Programm, 564
- Akzentbuchstaben eingeben, 57
- Alarmindikator, 51
- Algebra, 418–451
 - allgemeine Lösungen und Hauptlösungen, 426
 - Auswahlumgebung, 431
 - benutzerdefinierte Umformungen, 449
 - Beschränkungen, 422
 - Brüche addieren, 443
 - Gleichungen nach einer Variablen auflösen, 419
 - Isolieren einer Variablen, 421
 - Klammern erzeugen und bewegen, 437
 - mehrfaches Ausführen von Umformungen mit RULES, 444
 - Potenzen umformen, 442
 - Produkte und Potenzen erweitern, 429
 - quadratische Gleichungen lösen, 423

- symbolische Lösungen, 421
- Terme neuordnen, 427
- Terme verschieben, 436
- Terme zusammenfassen, 428, 436
- trigonometrische Funktionen erweitern, 443
- Umformungen mit RULES, 431
- universelle Umformungen, 434
- verdeckte Variablen anzeigen, 427
- Vergleich der Methoden zum Isolieren einer Variablen, 425
- Vertauschung, Verbindung, Verteilung, 438
- Algebraische Ausdrücke
 - Anwendung logischer Funktionen, 531
 - Anwendung von Vergleichsfunktionen, 530
 - Eingabemodus, 81
 - komplexe Zahlen, 177
 - Modus zur Eingabe in Programmen, 82
 - Zusammenfassen von Ausdrücken, 618
- algebraische Objekte, 134 – 139
 - algebraische Objekte im EquationWriter ersetzen, 267
 - auswerten, 134
 - Differentiation, 453
 - EquationWriter, 259
 - komplexe Zahlen, 174
 - mathematische Ausdrücke, 90
 - Neuordnen von Termen, 431
 - Objekte mit Einheiten, 207
 - Objekttypen-Nummer, 102
 - Priorität der Operatoren, 138
 - schrittweise Auswertung, 135
 - Terme auswerten, 138
 - trennen, 95
 - vereinfachen, 137
 - Vergleich mit Programmen, 134
 - verschachtelte Klammern, 138
- Algebraischer Eingabemodus, 81
 - algebraischer Eingabemodus Indikator, 51, 89
 - Objekte mit Einheiten eingeben, 203
- Algebraischer Eingabemodus, Objekte mit Einheiten eingeben, 205
- Algebraischer Programmeingabemodus, 82
- ALGEBRA-Menü, 421, 428
- allgemeine Lösungen, Gleichungen, 426
- Alpha Taste, 26
- Alpha-Arretierung, 242
- Alpha-Eingabemodus, 56, 242
- Alpha-Tastatur, Indikator für, 51
- Alpha-Tastenbelegung, 56
- Alpha-Tastenbelegung, links umgeschaltet, 52
- Alpha-Tastenbelegung, rechts umgeschaltet, 52
- ALRMDAT reservierte Variable, enthält die Daten für einen Termin, 115
- ALRM-Menü, 479
- AM/PM-Anzeige der Uhrzeit, 477
- Analytische Funktionen, 44
- AND, Binärzahlen, 227
- Ändern des Inhalts einer Variablen, 118
- Änderung des Vorzeichens, einer Zahl, 49
- Anhalten, Programme mit dem Befehl HALT, 565
- Anpassen des Taschenrechners, 242
 - Benutzertastatur, 234
 - Modus einstellen, 239
 - Umschalttasten, 233

- Verwendung von Systemflags, 241
- Antworten auf häufige Fragen, 712
- Anwender-Flags, 555
- Anzeigemodus
 - ändern, 63
 - Anzeigeformat für Zahlen, 61
- Anzeigen eines Objektes, aus einem Programm, 565
- Anzeigen verdeckter Variablen, 427
- Arcuscosinus, 150
- Arcussinus, 150
- Arcustangens, 150
- Areacosinus, 147
- Areasinus, 147
- Areatangens, 147
- Argumente im Stack, 65
- Arithmetik
 - Datumsarithmetik, 491
 - komplexe Felder, 386
 - komplexe Zahlen, 169
 - Matrix und Vektor, 384
 - mit Variablen, 123
 - Objekte mit Einheiten, 216
 - Vektoren, 382
 - Zeitarithmetik, 492
- Arithmetik und Grundrechenarten, 144 – 145
- ATTN-Taste, 25, 58
- Aufforderung zur Eingabe, 561
- Auflösung, 346
 - statistische Darstellungen, 347
 - Zeichengeschwindigkeit erhöhen, 347
- aufrufen
 - benutzerdefinierte Tastenbelegungen, 239
 - den Inhalt einer Variablen, 118
- Aufrufen, Flags, 558
- Ausdrücke
 - kein “=”-Zeichen, 138
 - komplexe Zahlen, 177
- ausführen
 - Befehle und Funktionen vom Stack, 65
 - benutzerdefinierte Funktionen, 164
- Ausgabe, 575
- Ausschalttaste, 25
- Austesten
 - Programme, 521
 - Unterprogramme, 524
- Auswahlumgebung, 262, 431
- Auswahl-Umgebung, Teilausdrücke bearbeiten, 263
- Auswerten
 - algebraische Objekte auswerten, 134, 135
- auswerten
 - des Inhalts einer Zeichenkette, 95
 - einer Variablen, 116
 - einer Variablen verhindern, 89
 - Lösungsmodus, 136
 - Priorität der Operatoren, 138
 - symbolische Konstanten, 156, 242
 - von Variablen, die Programme enthalten, 117
 - von Variablen durch Anführungszeichen verhindern, 119
- Auswerten von Variablen, Verzeichnisse nach dem Namen der V. absuchen, 130
- Auswertung, von lokalen Variablen, 513
- automatische Alpha-Arretierung, 242
- automatische Skalierung eines Diagramms, 319
- automatisches Abschalten nach 10 Minuten, 25

B

- Balkendiagramm, 410
 - PLOT, 363
 - Statistik, 409
- BASE-Menü, 87, 225, 227
- Basis
 - Binärzahlen, 224
 - wählen, 225
- Basiszeichen, 224
 - eingeben, 226
- Batterien, 25, 717
 - für den HP 48, 718
 - für RAM-Einsteckkarten, 693, 718
 - wechseln, 718
- Batterien austauschen, 717
- bearbeiten
 - Gleichungen im EquationWriter bearbeiten, 259
 - in der Befehlszeile, 80
- Bearbeitung mit der Rückschritttaste, EquationWriter, 260
- bedingte Strukturen
 - in Programmen, 532, 538
- Befehl, Definition, 44
- Befehle
 - einwertig, 65
 - für Objekte, 95
 - zweiwertig, 66
- Befehle im Kermit-Protokoll, 665
- Befehlsargumente aus dem Stack, 65
- Befehlszeile, 80–83
 - Bearbeitung im EquationWriter, 261
 - Felder eingeben, 377
 - löschen mit der ATTN-Taste, 58
 - mittlerer Abschnitt des Displays, 47
 - Text eingeben und bearbeiten, 48, 80
 - Zahlen eintippen, 49
 - Zurückholen der vorherigen Befehlszeile, 83
- Befehlszeilen-Zeichenkette, Aufbau, 570
- Begrenzungszeichen, 504
 - [] begrenzen Felder, 186, 374
 - () begrenzen komplexe Zahlen, 86
 - ' ' begrenzt algebraische Objekte, 90
 - [] begrenzt Felder, 88
 - () begrenzt komplexe Zahlen, 171
 - { } begrenzt Listen, 92
 - : : begrenzt markierte Objekte, 92
 - « » begrenzt Programme, 91
 - " " begrenzt Zeichenketten, 91
 - ' ' verhindern das Auswerten einer Variablen, 119
 - ' ' verhindert Auswertung einer Variablen, 89
 - # begrenzt Binärzahlen, 87
 - _ begrenzt Objekte mit Einheiten, 93, 202
 - = bezeichnet Gleichungen, 139
 - # leitet Binärzahlen ein, 224, 226
- Belegt-Indikator, 51
- benutzerdefinierte Ableitungen, 456
- benutzerdefinierte Einheiten, 222
- benutzerdefinierte Fehler, 591
- benutzerdefinierte Funktionen, 162–167
 - ausführen, 164
 - erzeugen, 164
 - plotten, 360
 - Programme, 167
 - Vergleich mit eingebauten Funktionen, 162
 - verschachteln, 165

- benutzerdefinierte Koordinaten in Zeichnungen, 349
- benutzerdefinierte Menüs, 231
 - Objekte mit Einheiten umwandeln, 211
- benutzerdefinierte Umformungen, 449
- Benutzermenüs
 - erzeugen, 231
 - Menüfelder, 231
 - umgeschaltete Tasten, 233
- Benutzermodus, 235, 242
- Benutzerspeicher, 106
- Benutzerspezifische Anpassung, 230
- Benutzertastatur, 234
 - Operationen zur Anpassung, 239
 - Reaktivieren einer Taste, 238
 - Tasten umbelegen, 235
 - Tastenbelegungen bearbeiten, 239
 - Umbelegung löschen, 238
- Bereich
 - reelle Zahlen, 86
 - Überschreitung (reelle Zahlen), 86
 - Unterschreitung (reelle Zahlen), 86
- beschriften, Plotachsen, 346
- Bessel-Funktionen, 634
- Bestimmte Schleifen, 540
- Bestimmte Schleifen
 - START...NEXT, 540
- Bestimmte Schleifen
 - START...STEP, 543
- Betrag, 159
 - komplexe Zahlen, 170, 179
 - Matrix, 388
 - Vektor, 190
 - Vektoren, 382
- Betriebsarten, zum Druck, 658
- bewegen, Stackzeiger, 77
- Bewegte Darstellung
 - einer benutzerspezifischen Grafik, 647
 - von Taylor-Polynomen, 637
- Bibliothekbefehle, 709
- Bibliothekobjekte, Anbinden an ein Inhaltsverzeichnis, 707
- Bibliothekobjekte, 706
 - Objekttypen-Nummer, 102
- binär in reell, Umwandlung, 227
- binäre Arithmetik, 224–229
- Binärzahlen, 87
 - angezeigte Bits, 225
 - Anzeige, 600
 - anzeigen, 226
 - Basis, 224
 - Berechnungen, 226
 - eingeben, 226
 - interne Darstellung, 225
 - Logikbefehle, 227
 - Objekttypen-Nummer, 102
 - Wortlänge, 224
- “blaue” Tasten, 26, 52
- Bogenmaß-Modus, 149
- Boolesche Logikbefehle, 227
- Bruch als Näherungswert für eine Zahl, 144
- Brüche
 - Addition von Brüchen mit ALGEBRA, 443
 - EquationWriter, 249
 - Funktionen zur Umwandlung in Brüche, 146
 - Genauigkeit des Ergebnisses bei der Umwandlung in Brüche, 146
- Buchstaben
 - Akzente eingeben, 57
 - Großbuchstaben, 56
 - Kleinbuchstaben, 56
- BYTES-Befehl, 107

C

- chi-quadrat-Test, 415
- CONIC-Plottyp, 353, 355
- Cosinus, 150
- Cotangens, benutzerdefinierte Funktion, 163
- CST, reservierte Variable für Benutzermenüs, 231
- CST reservierte Variable, Daten für benutzerdefinierte Menüs, 115
- CST-Menü, 231
 - Objekte mit Einheiten umwandeln, 211
- Cursortasten, 26

D

- Dateien, Senden und Empfangen, 665
- Dateien holen, Eingabe/Ausgabe, 665
- Dateinamen, PC im Vergleich mit dem HP 48, 681
- Daten empfangen, serielle Ein- und Ausgabe, 665
- Daten senden, serielle Ein- und Ausgabe, 665
- Datenausgabe, 575
 - Kennzeichnung mit Zeichenketten-Befehlen, 576
- Datum
 - Anzeige ändern, 477
 - Befehle, 476
 - Datum in Zahl umwandeln, 491
 - Datum in Zeichenkette umwandeln, 491
 - Datumsarithmetik, 491
 - Einstellen des Datums, 476
 - Monat/Tag/Jahr-Anzeige, 477
 - Tag/Monat/Jahr-Anzeige, 477
- Definieren, benutzerdefinierte

- Funktionen, 164
- definieren, von Variablen, 115
- dekadischer Antilogarithmus (Basis 10), 147
- dekadischer Logarithmus (Basis 10), 147
- Determinante, berechnen, 388
- Dezimalstellen, Zahl der angezeigten, 62
- Dezimalzeichen, 62
- Dialogprogramme
 - Akustisches Signal, 564
 - Anhalten von Programmen, 565
 - Anzeige eines internen Menüs, 578
 - Anzeigen von Objekten, 565
 - Aufforderung zur Dateneingabe, 566
 - Aufforderung zur Eingabe, 561
 - Ausgeben einer Tastenadresse, 584
 - Eine Befehlszeilen-Zeichenkette aufbauen, 570
 - Einfrieren eines Teils des Anzeigefeldes, 565
 - Erzeugung eines temporären Menüs, 583
 - Kennzeichnen einer Programmausgabe, 575
 - Verwenden von benutzerspezifischen Menüs, 579
 - Verwenden von markierten Objekten als Datenausgabe, 575
 - Verwenden von Zeichenketten-Befehlen zur Kennzeichnung ausgegebener Daten, 576
 - Wahlmöglichkeiten für den Eingabebefehl, 569
- Differentiation
 - algebraische Ausdrücke differenzieren, 453

- benutzerdefinierte Funktionen, 162
- Differentiation in einem Schritt, 456
- eingebaute Funktionen, 162
- schrittweise Differentiation, 454
- Differenz, Zahl der Tage zwischen zwei Daten, 491
- Direkteingabemodus, 81
 - Objekte mit Einheiten eingeben, 203
- Display, 47
 - Kontrast einstellen, 25
 - Meldungen löschen, 50
 - Statuszeilen, 50
- dividieren
 - eine Variable durch einen Wert aus dem Stack, 124
 - einen Vektor durch eine Matrix, 385
- EquationWriter, 249
- Zahlen, 144
- DOERROR,
 - Fehlervverzweigungen, 587
- Druck
 - Betriebsarten, 658
 - des Anzeigehalts, 655
 - des Stacks, 656
 - Einstellen der Verzögerung, 657
 - Escape-Sequenzen und Steuersequenzen, 658
 - Grafische Daten, 660
 - grafische Objekte, 657
 - PRTPAR* enthält
 - Druckparameter, 661
 - Sammeln von Daten im Pufferspeicher, 659
 - über den seriellen Anschluß, 660
 - Variable, 656
 - Zeichenketten, 656
 - Zeichensatz des HP 48, 658
 - zweizeilig, 657

- Druck aus dem Pufferspeicher, 659
- Drucken, 652–662
 - Schleifentest, 728
- Druckkopf-Rücklauf,
 - Speicherinhalt des Druckerpuffers ausgeben, 653
- Drucktakt verzögern, 653, 657
- Dualzahlen, 224
- durchgehende Diagramme, 323

E

- e*, eingebaute Konstante, 154
- E/A-Menü, 665, 685
- Ebene löschen, Stack, 68
- ebene Winkel, 214
- Ebenen des Stacks, 48
 - Nummer der aktuellen Stackebene ausgeben, 77
- Ebenen im Stack tauschen, 67
- EDIT-Menü, 73
- eindimensionale Vektoren, 88
- Einfügen, Zeile oder Spalte einer Matrix, 379
- Eingabe
 - Aufforderung zur Dateneingabe, 566
 - Wahlmöglichkeiten, 569
- Eingabe/Ausgabe
 - Befehle für, 665
 - Befehle für die serielle, 685
 - Binärmodus/ASCII-Modus, 682
 - E/A-Parameter einstellen, 669
 - Kabelverbindung, 674
 - Kermit-Protokoll zur Übertragung von Dateien, 663
 - Konfiguration Lokal/Lokal, 672
 - Konfiguration Lokal/Server, 673

- Testen des seriellen Anschlusses, 729
 - vom HP 48 zum HP 48, 665
 - vom HP 48 zum HP 48, 671
 - vom PC zum HP 48, 674, 676
 - Zeichencodes umsetzen, 679
 - Zulässige Datenarten, 664
- Eingabe/Ausgabe übersetzen, 669
- Eingabe/Ausgabe, 663–687
- Eingabemodi, Matrizen eingeben, 379
- Eingabemodus, vier Arten, 81
- eingebaute Befehle, 95
 - Objekttypen-Nummer, 102
- eingebaute Funktionen, 95
 - Objekttypen-Nummer, 102
 - Vergleich mit benutzerdefinierten Funktionen, 162
- eingebaute Konstanten, 154
- eingebaute Objekte mit Einheiten, 209
- Eingefrorener Teil des Anzeigefeldes, 565
- Einheit, Systemzeit als ganze Binärzahl, 493
- Einheiten
 - Objekte mit, 93
 - Objekte mit E. aus dem Stack zusammenfügen, 95
 - Objekte mit E. trennen, 95
- Einheiten für Winkel, 214
- Einheitsmatrix, berechnen, 388
- Einheitsvektor, 191
 - komplexe Zahlen, 179
- Einschalttaste, 25
 - ATTN-Taste, 58
- Einsteckkarten, 688
 - Einbau und Ausbau, 689
 - Grenzwerte für die Umgebungsbedingungen, 717
- RAM, 106
- ROM, 106
- Einstellung
 - E/A-Parameter einstellen, 665
 - Unterbrechung der seriellen Eingabe/Ausgabe, 685
- Einzelschritt
 - Ausführung eines Programms, 521
 - Programmoperationen, 521
- Empfang von Zeichenketten, Serielle Eingabe/Ausgabe, 685
- endliche Reihen, 458
- ENG-Modus, 62
- ENTER-Taste, 26
 - Ebene 1 kopieren, 70
- EquationWriter, 24, 244–269
 - Ableitungen, 252
 - Addition, Subtraktion und Multiplikation, 248
 - algebraische Objekte und Objekte mit Einheiten, 259
 - Aufbau, 246
 - Auswahl-Umgebung, 262
 - Bearbeitung in der Befehlszeile, 261
 - Bearbeitung mit der Rückschrittaste, 260
 - Division und Brüche, 249
 - Exponenten, 250
 - Funktion “wobei”, 255
 - Gleichungen bearbeiten, 259
 - Gleichungen erzeugen, 248
 - Integrale, 253
 - Klammern, 247
 - Klammern verwenden, 252, 255
 - Objekte aus dem Stack einfügen, 265
 - Objekte mit Einheiten, 254
 - Objekte mit Einheiten erzeugen, 220
 - Quadratwurzel und xte Wurzel, 251

- Summen, 253
- Tastatur, 247
- Teilausdrücke bearbeiten, 263
- Teilausdrücke ersetzen, 266
- Zahlen und Namen, 248
- Zehnerpotenzen, 252
- Ergebnisse im Stack, 65
- ersetzen, Teile einer Liste oder Zeichenkette, 95
- Erweitern von Produkten und Potenzen, 429
- Escape-Sequenzen, Druck, 658
- etc-Taste
 - Verwendung für die Eingabe von Akzentbuchstaben, 57
 - Verwendung zu Eingabe von Sonderzeichen, 57
- Euklidische Norm, berechnen, 388
- Exklusiv-ODER
 - bei Tests, 531
 - Binärzahlen, 227
- Exponent
 - Anzeigeformat, 62
 - anzeigen, 159
 - eintippen, 49
 - EquationWriter, 250
- Exponentialfunktionen, 147
- externe Objekte
 - Objekttypen-Nummer, 102
 - Speichern von Objekten in Speichererweiterungskarten, 94

F

- Fakultät, 158
- “Falsch”, Ergebnis eines Tests, 528
- FCN-Menü, 332
- Fehler
 - Befehle für
 - Fehlerverzweigungen, 587
 - benutzerdefiniert, 591
 - Fehlerbehandlung, 586

- Fehlermeldung, 587
- Fehlernummer, 587
- Fortsetzen des Programmablaufs, 587
- letzte Kermit-Fehlermeldung, 665
- Löschen des letzten, 587
- Fehlermeldungen, 50
- Felder, 88, 373–392
 - Befehle, 388
 - Dimensionen(Größe), 95
 - Eingabe über die Befehlszeile, 377
 - Element einfügen, 95
 - komplexe Felder, 386
 - zusammenfügen, 95
- Feuchtigkeit, Auswirkung auf den Taschenrechner, 717
- Fibonacci-Zahlen, 594
- FIX-Modus, 62
- Flächen, Flächeninhalt unter einer gezeichneten Kurve, 332
- Flag, Drucker, 660
- Flag für Drucker, 660
- Flag für E/A-Geräte, 660
- Flags, 555
 - Aufrufen und Speichern, 558, 602
 - Auswertung symbolischer Konstanten, 156
 - setzen, löschen und überprüfen, 241
 - Setzen, Rücksetzen und Testen, 556
 - Vollständige Liste der, 757
- formale Variable, 165
- Format
 - der gedruckten Datenausgabe, 654
 - von Zahlen im Display, 61
- Fortsetzen des Programmablaufs, nach einem Fehler, 587

freier Speicher, Zahl unbenutzter Bytes, 107
FUNCTION-Plottyp, 353, 354
Funktionen, Gleichungen als Argumente, 138
Funktion, Definition, 44
Funktion IFT, 538
Funktion IFTE, 539
Funktionen
 als Objekte, 95
 Analyse in der Grafikumgebung, 331
 benutzerdefinierte Funktionen, 162
 benutzerdefinierte Funktionen erzeugen, 164
 eingebaute Funktionen, 162
 mathematische, 142–161
 Tastatur, 144–145
 Verwenden symbolischer Argumente, 161
 Winkelumrechnung, 152
Funktionsargumente aus dem Stack, 65

G

ganzzahliger Teil, mathematische Funktion, 159
Garantie, 731
Genauigkeit
 der angezeigten Zahl, 62
 Lösen von Gleichungssystemen, 390
 π , 151
 Umwandlung in einen Bruch, 146
geometrische Reihen, 459
Gesicherten Anwenderspeicher rücladen, 678
Gleich, Vergleichsfunktion, 529
Gleichung ersten Grades, Gleichungen 1. Grades nach

x auflösen, 425
Gleichung im Stack, trennen, 95
Gleichungen
 allgemeine Lösungen und Hauptlösungen, 426
 Argumente einer Funktion, 138
 Erzeugung benutzerdefinierter Funktionen, 164
 Gleichungen im EquationWriter bearbeiten, 261
 Gleichungen mit PLOT lösen, 288
 Gleichungen nach einer Variablen auflösen, 419
 quadratische Gleichungen lösen, 421
 verbinden, 294
 “=”-Zeichen, 138
Gleichungskatalog, 274, 280
 Befehle, 281
 Gleichungen verbinden, 294
 Liste mit Gleichungen erzeugen, 296
 umsortieren, 281
 verlassen, 284
Gleichungssysteme, 385
 Genauigkeit der Lösungen, 390
 überbestimmte Gleichungssysteme, 392
 unterbestimmte Gleichungssysteme, 392
globale Namen, Objekttypen-Nummer, 102
globale Variable, 112
Gon-Modus, 149
Grad, Grad in Radiant umwandeln, 152
Grad-Modus, 149
 Indikator für, 51
Grafikobjekte
 Bearbeitung im Stack, 369
 Darstellung, 309
 Größe, 95

- Objekttypen-Nummer, 102
- Stackform, 367
- Verwendung in Programmen, 369
- Grafikumgebung, 324
 - gezeichnete Funktionen analysieren, 331
 - grafische Elemente zu *PICT* hinzufügen, 364
 - Vorstellung, 308
 - Zoomoperationen, 328
- Grafische Daten, Druck, 660
- Grafische Objekte, Druck, 657
- GRAPHICS-FCN-Menü, 332
- GRAPHICS-ZOOM-Menü, 329
- Graphikumgebung,
 - Stackoperationen in der Grafikumgebung, 368
- Grenzwerte für die Umgebungsbedingungen, Einsteckkarten, 717
- griechische Buchstaben, Eingabe über die Tastatur, 52
- Großbuchstaben, 52, 56
- Größe
 - einer Liste oder Zeichenkette, 95
 - eines Feldes, 95
 - eines Grafikobjektes, 95
 - von *PICT*, 351
- Größer als, Vergleichsfunktion, 529
- Größer als oder gleich, Vergleichsfunktion, 529
- größte ganze Zahl, mathematische Funktion, 159
- Grundgesamtheit, Statistik, 405
- Gültigkeitsbereich von lokalen Variablen, 513

H

- Häufigkeit, statistische Stichprobe, 404
- Hauptbelegung der Tastatur (nicht umgeschaltet), 52
- Hauptlösungen, Gleichungen, 426
- Hexadezimalzahlen, 87, 224
- Histogramm
 - PLOT, 363
 - Statistik, 409
- Histogramme, Statistik, 412
- HMS-Format, 492
- HOME*, Startverzeichnis, 50
- HOME*-Verzeichnis, 126, 131, 133
- HP Solve, 24, 270–304
 - aktuelle Gleichung angeben, 275
 - Arbeitsweise, 298
 - Ausdrücke lösen, 275
 - Ergebnisse überprüfen, 287
 - Gleichung aus dem Gleichungskatalog auswählen, 280
 - Gleichungen aufrufen, 277
 - Gleichungen bearbeiten, 277
 - Gleichungen lösen, 275, 277
 - Gleichungen speichern, 277
 - Interpretation der Lösungen, 301
 - keine Lösung gefunden, 304
 - Lösungen, 288
 - Lösungen für Programme ermitteln, 297
 - mehrere Lösungen, 288
 - Menü SOLVR anpassen, 291
 - Näherungen bestimmen, 288
 - neue Gleichung eingeben, 278
 - Objekte mit Einheiten verwenden, 289
 - Programme lösen, 275
 - Verwendung mit PLOT, 272
 - Vorzeichenwechsel, 302
- Hyperbelfunktionen, 147

hyperbolischer Cosinus, 147
hyperbolischer Sinus, 147
hyperbolischer Tangens, 147
HYP-Menü, 147

I

i (imaginäre Einheit), 154, 178
IFERR...THEN...ELSE...END
 Fehlerverzweigung für
 Programme, 590
IFERR...THEN...END
 Fehlerverzweigung für
 Programme, 588
Imaginärteil, 179
 komplexes Feld, 386
Indikator, vollständige Liste der
 Indikatoren, 51
Indikator für die
 Programmeingabe, 506
Infinitesimalrechnung, 452–471
 Differentiation, 453
 Differentiation
 benutzerdefinierter
 Funktionen, 456
 numerische Integration, 467
 Summen, 458
 symbolische Integration, 462
 Taylorpolynome als
 Näherungen, 466
 vollständige Differentiation, 456
 Wie der HP 48 symbolisch
 integriert, 464
Infrarot-Drucker, 652, 659
Infrarotdrucker, Schleifentest, 728
Infrarot-Drucker, Zeichensätze,
 658, 660
INSUFFICIENT MEMORY,
 Fehlermeldung, 109
Integrale
 EquationWriter, 253
 Integrale in die Befehlszeile
 eintippen, 462

Integrand, Näherung, 466
Integration
 Genauigkeitsfaktor, 468
 mit dem Stack integrieren, 471
 numerische Integration, 467
 symbolische Integration, 462
 Wie der HP 48 integriert, 464
Interaktive Programme, 560–585
Interaktiver Stack, 75–80
 aktivieren, 75
 Objekte darin anschauen, 78
 Operationen, 76
 verlassen, 79
Internationales Einheitensystem
 (SI), 202
interne Darstellung
 Binärzahlen, 225
 Vektoren, 185
Internes Menü, Anzeige, 578
Interpunktionszeichen, als
 Begrenzungszeichen, 58
inverse Matrix, 383
inverser hyperbolischer Sinus, 147
inverser hyperbolischer Tangens,
 147
IOPAR reservierte Variable,
 enthält E/A-Parameter, 115
Isolieren einer Variablen, Algebra,
 421
iteratives Verfahren, Lösen von
 Gleichungssystemen, 390

K

Kabelverbindung, vom PC zum
 HP 48, 674
Katalog, REVIEW, 120
Kataloge
 Gleichungskatalog, 274, 280
 Statistik-Katalog, 400
 Terminkatalog, 485
Kegelschnitte plotten, 355

- Kehrwert
 - Kehrwert einer Zahl, 144
 - Objekt mit Einheiten, 218
- Kennzeichnung
 - Datenausgabe mit Zeichenketten-Befehlen, 576
 - Programmausgabe, 575
- Kermit-Modus, Server/Lokal, 668
- Kermit-Protokoll zur Übertragung von Dateien, 663
- Kettenrechnungen mit dem Stack, 67
- Klammern
 - Algebra, 437
 - Eingabe von Vektoren, 186
 - EquationWriter, 255
 - höchste Priorität bei algebraischen Objekten, 138
 - komplexe Zahlen, 171
 - Verwendung im EquationWriter, 247, 252
- Kleinbuchstaben, 52, 56
- Kleiner als, Vergleichsfunktion, 529
- Kleiner als oder gleich, Vergleichsfunktion, 529
- kleinste ganze Zahl, mathematische Funktion, 159
- Kombinationen, berechnen, 158
- Komma, als Dezimalzeichen, 62
- Kompaktformat, der gedruckten Datenausgabe, 654
- Komplement, Binärzahl, 227
- komplex in reell, trennen, 95
- komplexe Felder
 - Arithmetik, 386
 - Befehle, 386
 - Objekttypen-Nummer, 102
- komplexe Zahlen, 86, 168–182
 - algebraische Objekte, 174
 - Anzeigeformat, 170
 - Arithmetik, 169
 - Ausdrücke, 177
 - Befehle, 179
 - Druck, 654
 - eingeben, 171
 - Ergebnis von Operationen mit reellen Zahlen, 176
 - Felder komplexer Zahlen, 386
 - i (imaginäre Einheit), 178
 - interne Darstellung, 170
 - konjugieren, 179
 - Objekttypen-Nummer, 102
 - trennen, 95
 - Umwandlung in reelle Zahlen, 179
 - Vergleich mit reellen Zahlen, 174
 - Vergleich mit Vektoren, 179, 180, 199
 - Winkelmodus ändern, 170
 - zerlegen, 173, 179
 - zusammensetzen, 173, 179
- Konfiguration Lokal/Lokal vom HP 48 zum HP 48, 672 vom PC zum HP 48, 676
- Konfiguration Lokal/Server vom HP 48 zum HP 48, 673 vom PC zum HP 48, 677
- konjugieren
 - des Inhalts einer Variablen, 123
 - komplexe Felder, 386
 - komplexe Zahlen, 179
- Konstanten, symbolische, 154
- Konstantenmatrix, berechnen, 388
- Kontrast
 - dunkler, 25
 - einstellen, 25
 - heller, 25
- Konvergenz, Untersuchen einer Reihe, 459
- Konvertierung, Objekte in eine Zeichenkette, 600
- Koordinatenmodus, ändern, 185

Koordinatensysteme für
 Zeichnungen, 349
kopieren
 Ebene 1 im Stack, 70
 Objekte im Stack, 83
Kopieren des Stacks, 83
Kopieren von Variablennamen,
 130
Korrektur von Tippfehlern, 49
Korrelation, 406
Kovarianz, 405, 406
Kreis, plotten, 364
Kreuzprodukt, 190, 382
Kugelkoordinaten-Modus,
 Indikator für, 51
Kugelmodus, 184
Kundenbetreuung, 712
Kundenspezifische Menüs, in
 Programmen, 579

L

LAST ARG, Zurückholen
 gelöschter Variablen, 123
LAST MENU-Taste, 61
Letzte CMD-Taste, 83
letztes Argument
 wiederverwenden, 68
lineare Gleichungen, 385
 Genauigkeit der Lösungen, 390
lineare Regression, 406
Linie, plotten, 364
linke Umschalttaste, 55
Listen
 eine Untermenge bilden, 95
 einen Teil ersetzen, 95
 Eingabemodus, 82
 Objekttypen-Nummer, 102
 Position eines Objektes, 95
 Zahl der Elemente (Größe), 95
 zusammenfügen, 95
Logarithmusfunktionen, 147
Logikbefehle, 227

Logische Funktionen, 531
 in algebraischen Ausdrücken,
 531
lokale Namen, Objekttypen-
 Nummer, 102
Lokale Variablen, 112
 Auswertung, 513
 Definitionsbereich, 513
 verwendet in Programmen, 510
Lokaler Modus, 668
löschen
 alle Variable in einem
 Verzeichnis, 123
 benutzerdefinierte
 Tastenbelegungen, 238
 Flags, 241
Löschen, letzter Fehler, 587
löschen, Markierung von
 markierten Objekten, 95
 Meldungen aus dem Display, 50
 PICT, 315, 348
Löschen, Sicherungsobjekte, 701
löschen, Stack, 68
 Termine löschen, 486
 Variable, 122
 Verzeichnisse löschen, 131
 von Objekten bei zu wenig
 Speicherplatz, 110
 Zeile oder Spalte einer Matrix,
 379
Löschen des gesamten Speichers,
 108
löschen mit ATTN, 25
lösen
 Gleichungssysteme, 385
 nach einer Variablen auflösen,
 421
 quadratische Gleichungen lösen,
 421, 423
Lösungsalgorithmus
 Anfangsnäherungen, 299
 HP Solve, 298
 unterbrechen, 299

Zwischennäherungen, 300

M

Mantisse

- Anzeigeformat, 62
- einer Zahl anzeigen, 159
- eintippen, 49

markierte Objekte, 92, 93

- als Datenausgabe, 575
- aus dem Stack zusammenfügen, 95

Markierung löschen, 95

Objekttypen-Nummer, 102

trennen, 95

mathematische Funktionen, 142–161

Vektoren, 190

Matrix, Druck, 654

MATRIX-Menü, 374

MatrixWriter, 374

- Eingabemodi zum Eingeben von Matrizen, 379

Felder eingeben, 379

statistische Daten eingeben, 399

Zeile oder Spalte löschen, 379

Zeile oder Spalten einfügen, 379

Matrizen, 88

addieren und subtrahieren, 383

Arithmetik mit Vektoren, 384

bearbeiten, 379

Befehle, 388

Determinante, 388

einen Vektor durch eine Matrix dividieren, 385

Einheitsmatrix, 388

eintippen, 374

Felder, 373

Inverse, 383

komplexe Matrizen, 386

Multiplikation, 383

Multiplikation mit einer Zahl, 383

neu dimensionieren, 388

transponieren, 388

MATR-Menü, 388

Maximum

mathematische Funktion, 159

Stichprobe, 404

MAXR, eingebaute Konstante, 154

Medianwert

einer Liste, 610

von statistischen Daten, 607

Mehrzeilenformat, der gedruckten Datenausgabe, 654

Meldungen, 735–751

aus dem Display löschen, 50

MEMORY-Arithmetic-Menü, 123

MEMORY-Menü, 107

Menü BRCH, 532, 540

Menü CTRL, 521

Menü I/O SETUP, 669

Menü LIBRARY, 707

Menü PRG BRCH, 532, 540

Menü PRG CTRL, 521

Menü PRG TEST, 529

Menü SETUP, 669

Menü TEST, 529

Menü-Beschreibung, EDIT, 73

Menübeschreibungen

ALGEBRA, 421, 428

CST, 231

E/A, 685

GRAPHICS FCN, 332

GRAPHICS ZOOM, 329

I/O SETUP, 669

LIBRARY, 707

MATRIX, 374

MEMORY, 107

MEMORY Arithmetic, 123

MODES, 61, 239

MTH, 143

MTH BASE, 87, 225, 227

MTH HYP, 147

MTH MATR, 388

- MTH PARTS, 148
- MTH PROB, 158, 415
- MTH VECTR, 152, 185, 198
- PLOT, 312
- PLOTR, 315
- PRG BRCH, 532, 540
- PRG CTRL, 521
- PRG OBJ, 95
- PRG STK, 83
- PRG TEST, 529
- Menü-Beschreibungen, PRINT, 653
- Menübeschreibungen, SOLVE, 274, 277
 - SOLVE SOLVR, 274, 285
 - STAT, 396
 - STAT MODL, 406
 - TIME, 475
 - TIME ADJST, 478
 - TIME ALRM, 479
 - TIME RPT, 481
 - TIME SET, 476
 - UNITS-Befehle, 202
 - UNITS-Katalog, 202, 209
 - UNITS-Katalogmenü, 203
 - VAR, 113, 120, 126
- Menübeschreibungen E/A, 665
- Menüfelder
 - Benutzermenüs, 231
 - Menütasten, 47, 59
 - Namen von Variablen, 113, 115
 - Strich zeigt Verzeichnis an, 126
- Menüs
 - Definieren von Menütasten, 59
 - durch mehrere Seiten blättern, 59
 - ins vorherige Menü wechseln, 61
 - nächstes und vorheriges Menü wählen, 59
 - Strich zeigt Untermenü an, 59
 - verlassen, 60
 - Verwendung in Programmen, 578
- wählen, 60
- Menütasten, 59
- Minimum
 - mathematische Funktion, 159
 - Stichprobe, 404
- MINR, eingebaute Konstante, 154
- Mittelwert, Stichprobe, 404
- MOD (modulo), mathematische Funktion, 159
- Modell, Statistik, 406
- MODES, Menü, 239
- MODES-Menü, 61
- Modi
 - Rücksetzen durch Löschen des Speichers, 108
 - Server, 665
- MODL-Menü, 406
- Modus
 - algebraischer Eingabemodus, 81, 205
 - algebraischer
 - Programmeingabe-Modus, 82
 - Alpha-Eingabemodus, 56, 242
 - ändern, 600
 - Auswerten symbolischer Konstanten, 242
 - automatische Alpha-Arretierung, 242
 - Benutzermodus, 235, 242
 - Binärübertragung, 682
 - Bogenmaß, 149
 - Direkteingabemodus, 81
 - einstellen, 61
 - Festkommamodus, 62
 - Gon, 149
 - Grad, 149
 - Koordinatenmodus ändern, 185
 - Kugelmodus, 184
 - Modus durch Systemflags einstellen, 241
 - numerische Lösungen, 136, 155

- Polarkoordinaten, 86, 170
- Polarmodus, 184
- Programmeingabemodus, 82
- Rechteckkoordinaten, 86, 170
- Rechteckmodus, 184
- Standardmodus, 62
- symbolische Lösungen, 136, 155
- technischer, 62
- wählen, 239
- wissenschaftlicher, 62
- Zylindermodus, 184
- Modus ASCII-Übertragung, 669, 682
- Modus Binärübertragung, 669
- Modus Binär-Übertragung, 682
- Modus Infrarot-Übertragung, 669
- Modus Leitungsübertragung, 669
- Modusarten
 - ASCII-Übertragung, 682
 - Lokal, 668
 - Programmeingabe, 506
 - Server, 668
- Monat/Tag/Jahr-Anzeige, 477
- MTH BASE-Menü, 227
- MTH HYP-Menü, 147
- MTH PARTS-Menü, 148
- MTH PROB-Menü, 158
- MTH VECTR-Menü, 152, 185, 198
- MTH-BASE-Menü, 87, 225
- MTH-MATR-Menü, 388
- MTH-Menü, 143
- MTH-PROB-Menü, 415
- Multiplikation
 - Matrizen, 383
 - Matrizen mit einer Zahl multiplizieren, 383
- multiplizieren
 - eine Variable mit einem Wert aus dem Stack, 123
 - EquationWriter, 248
 - Zahlen, 144

N

- Näherungswert
 - Näherungswert für das bestimmte Integral, 467
 - symbolische Konstante, 155
- Namen
 - EquationWriter, 248
 - für Variablen, 89
 - in Variablen, 118
 - Namen von Einheiten auflisten, 206
- natürlicher Antilogarithmus (Basis e), 147
- natürlicher Logarithmus (Basis e), 147
- negative Zahl, komplexes Argument, 179
- negative Zahlen, eintippen, 49
- negieren, des Inhalts einer Variablen, 123
- Neudimensionieren einer Matrix, 388
- Neufestlegen von Terminen, 483
- Neuordnung, des Menüs VAR, 121
- NICHT, bei Tests, 531
- nicht durchgehende Diagramme, 323
- NO ROOM FOR LAST STACK, Fehlermeldung, 109
- NO ROOM TO SHOW STACK, Fehlermeldung, 109
- normale Variable, 112
- Normalverteilung, 416
- NOT, Binärzahlen, 227
- Nullstelle, geplottete Funktion, 332
- numerische Integration, 467
 - Genauigkeitsfaktor, 468
- numerische Konstanten, 154
- numerische Lösungen
 - Modus, 155

Modus für numerische
Lösungen, 136
numerischer Wert eines Zeichens,
95
NXT-Taste, wählt nächstes Menü,
59

O

obere Wahrscheinlichkeiten, 415
Objekt, in Zeichenkette
umwandeln, 95
Objekte
anzeigen und bearbeiten, 70, 71
Befehle zur Bearbeitung, 95
im Interaktiven Stack
anschauen, 78
Objekte aus dem Stack in den
EquationWriter einfügen,
265
Prüfsumme, 107
trennen, 95
Objekte mit Einheiten
Benutzermenüs, 231
EquationWriter, 254, 259
HP Solve, 289
Objekttypen-Nummer, 102
plotten, 361
Syntax, 202
Objekttypen, 102
abgezählte Zeichenketten, 91
Bibliothekobjekte, 94
Binärzahlen, 87
eingebaute Befehle, 95
eingebaute Funktionen, 95
Felder, 88
Grafikobjekte, 92
komplexe Zahlen, 86
Listen, 92
markierte Objekte, 92
Matrizen, 88
Namen, 89
Objekte mit Einheiten, 93

Objekttypen-Nummer, 102
Programme, 91
reelle Zahlen, 86
Sicherungsobjekte, 94
Vektoren, 88
Verzeichnisse, 94
XLIB-Namen, 94
Zeichenketten, 91
Objekttypen-Nummer, 102
bestimmen, 102
Objekttypen, 85
OBJ-Menü, 95
ODER, bei Tests, 531
Oktalzahlen, 87, 224
Operation, Definition, 44
OR, Binärzahlen, 227
"orangefarbene" Tasten, 26, 52
OUT OF MEMORY,
Fehlermeldung, 110

P

π , eingebaute Konstante, 154
Paket, Befehle zu einem Server
senden, 665
Pakete, Befehle an einen Server
senden, 684
PARAMETRIC-Plottyp, 353, 358
Parität
Einstellen, 671
Einstellung, 669
während des Drucks, 660
PARTS-Menü, 148
PATH, Ausgeben des aktuellen
Verzeichnispfads, 128
PC-Dateinamen im Vergleich mit
HP 48-Dateinamen, 681
Permanentspeicher, 25
Permutationen, berechnen, 158
photometrische Einheiten, 215
pi, 151
PICT
Bearbeitung im Stack, 368

- Größe ändern, 351
- Hinzufügen grafischer Elemente, 363
- löschen, 315
- löschen und auf die Grundeinstellung zurücksetzen, 348
- Piepston, Akustisches Signal abschalten bei Terminen, 483
- Pixelkoordinaten in Zeichnungen, 349
- Plot, 24
- PLOT
 - Datenelemente, 308
 - Struktur, 308
 - Verwendung mit HP Solve, 272
 - zwei Menüs und eine spezielle Umgebung, 308
- PLOT PLOTTR-Menü, 315
- Plotbereich
 - Plotbereich festlegen, 345
 - Plottypen PARAMETRIC und TRUTH, 345
- PLOT-Menü, 312
- plotten, 305 – 363
 - Ableitung einer geplotteten Funktion, 332
 - Angaben des Mittelpunktes und der Skalierung, 318
 - Arbeit mit schwierigen Diagrammen, 339
 - automatische Skalierung, 319
 - durchgehende und nicht durchgehende Diagramme, 323
 - Funktionen, 354
 - gezeichnete Funktionen analysieren, 331
 - Kegelschnitte, 355
 - Objekte mit Einheiten, 361
 - Plotparameter festlegen, 314
 - Plottyp PARAMETRIC, 358
 - polare Darstellungen, 357
 - Programme und benutzerdefinierte Funktionen, 360
 - Rechteck vergrößern, 330
 - statistische Daten in PLOT, 363
 - Statusmeldung gibt Plotparameter an, 314
 - TRUTH-Plottyp, 359
 - unabhängige Variable angeben, 317
 - was der HP 48 darstellen kann, 311
 - wie DRAW Punkte zeichnet, 322
 - x-Achse, 318
 - y-Achse, 318
 - Zoomoperationen, 328
 - Zurücksetzen der Plotparameter, 315
 - zwei oder mehr Gleichungen, 323
- Plottypen, 353
 - BAR, 354, 363
 - CONIC, 353, 355
 - FUNCTION, 353, 354
 - HISTOGRAM, 354, 363
 - PARAMETRIC, 353, 358
 - POLAR, 353, 357
 - SCATTER, 354, 363
 - TRUTH, 353, 359
- polarer Winkel, 179
- Polarkoordinaten, 86
- Polarmodus, 170, 184
 - Indikator, 170, 184
- POLAR-Plottyp, 353, 357
- Position eines Objektes in einer Liste, 95
- Potenzen, Potenzen mit Algebra umformen, 442
- PPAR reservierte Variable, enthält Parameter für PLOT, 115

- PPAR, reservierte Variable, Plotparameter, 347
- PREV-Taste
 - umschalten rechts führt zur ersten Seite, 59
 - wählt vorheriges Menü, 59
- PRG OBJ-Menü, 95
- PRG STK-Menü, 83
- Primärtastatur (nicht umgeschaltet), 26
- PRINT-Menü, 653
- Priorität von Funktionen
 - algebraische Objekte, 138
 - Objekte mit Einheiten, 207
- PROB-Menü, 158, 415
- Produkte und Potenzen, Erweitern von Produkten und Potenzen, 429
- Programmablauf, Fortsetzen des Programms nach einem Fehler, 587
- Programmablauf fortsetzen, 521
- Programme
 - Abbruch mit der ATTN-Taste, 58
 - Ablauf, 508
 - Ablauf fortsetzen, 521
 - als Argumente, 616
 - Anwendung von Tests, 528
 - Aufbau von Programmschleifen, 540
 - bedingte Strukturen, 532
 - Befehle zur Dateneingabe, 561
 - Berechnung der Ausführungszeit, 597
 - die mit Daten im Stack arbeiten, 516
 - die wie benutzerdefinierte Funktionen wirken, 516
 - Editieren, 509
 - Eingabe, 506
 - Eingabe/Ausgabe, 560
 - Eingabemodus, 82
 - Einzelstrettausführung, 521
 - Gültigkeitsbereich von lokalen Variablen, 513
 - lokale Namen auswerten, 513
 - Lösungen mit HP Solve ermitteln, 297
 - Objekttypen-Nummer, 102
 - plotten, 360
 - Struktur CASE...END, 536
 - Struktur DO...UNTIL...END, 549
 - Struktur FOR...NEXT, 545
 - Struktur FOR...STEP, 547
 - Struktur
 - IF...THEN...ELSE...END, 534
 - Struktur IF...THEN...END, 532
 - Struktur START...NEXT, 540
 - Struktur START...STEP, 543
 - Struktur
 - WHILE...REPEAT...END, 551
- Unterbrechen des Ablaufs mit dem Befehl WAIT, 578
- Unterbrechung, 521
- Variablen, die Programme enthalten, auswerten, 117
- Vergleich mit algebraischen Objekten, 134
- Verwenden von Grafikobjekten, 369
- Verwenden von kundenspezifischen Menüs, 579
- Verwenden von lokalen Variablen in, 510
- Verwenden von Terminen in Programmen, 489
- Verwendung durch andere Programme, 631
- Verwendung von Unterprogrammen in, 518

Programmeingabe-Modus, 82, 506
 Indikator für, 51
 Objekte mit Einheiten
 eingeben, 203
Programmierbeispiele, 593 – 649
Programmstruktur CASE...END,
 536
Programmstruktur
 DO...UNTIL...END, 549
Prozent des Gesamtbetrages,
 berechnen, 148
Prozentrechnen, Objekte mit
 Einheiten, 219
prozentuale Änderung, berechnen,
 148
PRTPAR reservierte Variable,
 enthält Druck-Parameter,
 115
Prüfsumme, 107, 593
 bei Eingabe/Ausgabe, 669
Pseudo-Zufallszahlen, 158
Pufferlänge, Serielle
 Eingabe/Ausgabe, 685
Puffern von Tastenfolgen, 51
Punkt, als Dezimalzeichen, 62

Q

quadratische Gleichungen
 Lösen quadratischer
 Gleichungen, 421, 423
quadratische Matrix, Inverse, 383
Quadratwurzel
 EquationWriter, 251
 Quadratwurzel einer Zahl, 144
Quadrieren einer Zahl, 144

R

Radiant, Radiant in Grad
 umwandeln, 152
Radiant-Modus, 149
 Indikator für, 51

RAM
 Benutzerspeicher, 107
 erweiterbar, 106
 (Schreib-Lese-Speicher), 688
RAM-Karten, 693
 Batterien, 693
 Einbau und Ausbau, 689
 Erweiterung des
 Benutzerspeichers, 697
 Schreibschutzschalter, 695
 Verwendung für die
 Datensicherung, 699
räumliche Winkel, 214
Realteil
 komplexe Zahl, 179
 komplexes Feld, 386
rechte Umschalttaste, 55
Rechteck, plotten, 364
Rechteckkoordinaten, 86
Rechteckmodus, 170, 184
rechts umgeschaltete
 Tastenbelegung, 52
reell in komplex, zusammenfügen,
 95
reelle Felder, Objekttypen-
 Nummer, 102
reelle Zahlen, 86
 Anzeigeformat, 62
 Bereich, 86
 Bereichsüberschreitung, 86
 Bereichsunterschreitung, 86
 größtmögliche, 86
 kleinstmögliche, 86
 MAXR und MINR, 154
 Objekttypen-Nummer, 102
reelle Zahlen in Brüche
 umwandeln, 146
 Umwandlung in reelle Zahlen,
 179
 Vergleich mit komplexen
 Zahlen, 174
Regeln der Priorität, algebraische
 Objekte, 138

Register, Variable stattdessen verwenden, 112

Rekursion, Berechnung von Fibonacci-Zahlen, 594

Reparatur, 732

reservierte Variable, 115

Reservierte Variable *IOPAR*, speichert E/A-Parameter, 670

Reservierte Variable *PRTPAR*, enthält Druckparameter, 661

REVIEW-Katalog, 120

Rollen
 der Befehlszeile, 48
 Stack, 70

Rollen des Stacks, 83

ROM
 erweiterbar, 106
 (Festwertspeicher), 688

ROM-Karten, Einbau und Ausbau, 689

ROM-Steckkarten, 706

Rotate-Befehle, Binärzahlen, 227

RPT-Menü, 481

Rückschritttaste, Verwendung in der Befehlszeile, 80

Rücksetzen, Flags, 556

Rückwärtszählen,
 Programmschleifenzähler, 553

RULES-Umformungen, 431 – 451
 Aktivieren der Umformungen mit RULES, 433
 Ausführen einer Umformung, 433
 Beispiele, 434
 Verlassen eines RULES-Menüs, 433

Runden von Zahlen, 159

Rundungsfehler, Lösen von Gleichungssystemen, 390

S

Schleife FOR...NEXT, 545

Schleife FOR...STEP, 547

Schleifen, 540
 DO...UNTIL...END, 549
 FOR...NEXT, 545
 FOR...STEP, 547
 START...NEXT, 540
 START...STEP, 543
 Wert des Schleifenzählers erhöhen, 553
 Wert des Schleifenzählers verringern, 553

WHILE...REPEAT...END, 551

Schleifentest, Infrarotdrucker, 728

Schreibschutzschalter
 Einbau von Einsteckkarten, 689
 in RAM-Karten, 695

schrittweise Differentiation, 454

schwache Batterie (Alarm),
 Indikator für, 51

SCI-Modus, 62

Σ DAT reservierte Variable, enthält die aktuelle statistische Matrix, 115

Σ DAT, reservierte Variable, enthält die aktuelle statistische Matrix, 398

Selbsttest, 724

Senden, Serielle
 Eingabe/Ausgabe, 685

Senden einer seriellen
 Unterbrechung, 685

Serielle E/A-Befehle, 685

Seriellen Anschluß öffnen, 665

Seriellen Anschluß schließen, 665

Serieller Anschluß
 Druck, 660
 Konfigurieren für den Druck, 660
 öffnen und schließen, 665

- Serielles Kabel, vom PC zum HP
 - 48, 674
- Servermodus, 665, 668
 - Starten und Beenden, 665
- Servermodus beenden, 665
- Service, 732
- SET-Menü, 476
- setzen, Flags, 241
- Shift-Befehle, Binärzahlen, 227
- Sichern von Inhaltsverzeichnissen, 700
- Sicherungsobjekte, 699, 700
 - Benutzermenüs, 232
- signifikante Stellen, 62
- signifikanteste Bits, Binärzahlen, 225
- Sinus, 150
- Skalarprodukt, 190, 382
- Skalieren einer Zeichnung, 318
- Snedecorsche F-Verteilung, 415
- SOLVE SOLVR-Menü, 285
- SOLVE-Menü, 274, 277
- Solver-Liste, benennen, 292
- SOLVR-Menü, 274, 285
 - anpassen, 291
- Sonderzeichen
 - Eingabe über die Tastatur, 52
 - Tabelle, 57
- Spalten-Norm, Matrix, 388
- Σ PAR reservierte Variable, enthält
 - Parameter für STAT, 115
- Σ PAR, reservierte Variable,
 - Statistik-Parameter, 408
- Speicher, 25
 - archivieren, 703
 - Archivierung, 677
 - eingebundenen Speicher freigeben, 704
 - erweitern, 106
 - gesicherten Anwenderspeicher rücladen, 678
- INSUFFICIENT MEMORY, 109
 - löschen, 108
 - Löschvorgang abbrechen, 108
 - Menge, die Objekte belegen, 107
 - NO ROOM FOR LAST STACK, 109
 - NO ROOM TO SHOW STACK, 109
 - OUT OF MEMORY, 110
 - Prüfsumme eines Objektes, 107
 - RAM und ROM, 106, 688
 - Sicherung, 677
 - Zahl unbenutzer Bytes, 107
 - zu wenig Speicherplatz, 109
- Speicher archivieren, 703
- Speicher freigeben, 704
- Speicher zurücksetzen, 108
- Speicherebenen, Stack, 64
- Speicherinhalt archivieren, 677
- Speichern, Flags, 558
- speichern
 - umbelegte Tasten, 235
 - Variablen, 114
- Speicherplätze, der Stack, 48
- Stack
 - Anschauen und Bearbeiten von Objekten, 71
 - Anschauen und Bearbeiten von Variablen, 72
 - Befehle, 83
 - Ebene 1 im Stack, 70
 - Ebene 1 in den EquationWriter einfügen, 265
 - Ebene 1 kopieren, 76
 - Ebenen, 48
 - Ebenen 1 und 2 tauschen, 67
 - eine Ebene löschen, 68
 - einwertige Befehle, 65
 - Gleichungen trennen, 95
 - letztes Argument wiederverwenden, 68
 - löschen, 68, 76
 - mehrere Speicherebenen, 64

- NO ROOM TO SHOW
- STACK, 109
- normale Berechnungen, 65
- Operationen in der
 - Grafikumgebung, 368
- Rechnen mit vorherigen
 - Ergebnissen, 67
- speichert Grafikobjekte, 92
- zweiwertige Befehle, 66
- Stack zu Einheit, zusammenfügen, 95
- Stack zu Liste, trennen, 95
- Stack zu Markierung, umwandeln, 95
- Stack zum Feld, zusammenfügen, 95
- Stackanzeige, 47
- Stackinhalt betrachten, 76
- Stackinhalt herausnehmen, 76
- Stackinhalt kopieren, 76
- Stackinhalt rollen, 76
- Stacks, Druck, 656
- Stackzeiger, bewegen, 77
- Stammfunktion, 462
- Standardabweichung, 404, 405
- Standardmodus, 62
- Startverzeichnis, *HOME*, 50
- Statistik
 - abhängige Variable, 406
 - Befehle für Summen, 414
 - Daten bearbeiten, 397, 400
 - Daten eingeben, 397, 399
 - Festlegen der aktuellen Matrix, 398
 - Grundgesamtheit, 405
 - Statistik mit Stichproben, 404
 - Stichproben zeichnen, 409
 - unabhängige Variable, 406
 - verbundene Stichproben, 406
- STATISTIK, 393
- Statistik, 417
- Statistik-Katalog, 400
 - Operationen, 401
 - umsortieren, 401
- STAT-Menü, 396
- STAT-MODL-Menü, 406
- Statuszeilen
 - Anzeigen des aktuellen Pfads, 127
 - auf dem Display, 50
- Steigung, gezeichnete Funktion, 332
- Stellvertreterzeichen, bei
 - Sicherungsobjekten, 702
- Steuercodes, Druck, 658
- Steuerungstermine, eingeben, 484
- Stichproben, 404
- STK-Menü, 83
- Störsicherheit, 733
- Streudiagramm
 - PLOT, 363
 - Statistik, 409
- Strich über Menüfeld, 126
- Struktur
 - IF...THEN...ELSE...END für Programme, 534
- Struktur IF...THEN...END für Programme, 532
- Studentsche t-Verteilung, 415
- subtrahieren
 - einen Wert aus dem Stack von einer Variablen, 123
 - EquationWriter, 248
 - Zahlen, 144
- Summe, Stichprobe, 404
- Summen, 458
 - Berechnen von Summen mit dem Stack, 460
 - Eingeben von Summen, 458
 - EquationWriter, 253
 - Statistik, 414
- symbolische Argumente,
 - Verwendung für Funktionen, 161
- symbolische Berechnungen, 24
- symbolische Integration, 462

- symbolische Konstanten, 154
 - auswerten, 156
 - e , 154
 - i (die imaginäre Einheit), 154, 178
 - π , 154
 - π , 151
 - Zahlenwerte, 155
- symbolische Lösungen
 - Modus, 155
 - Modus für symbolische Lösungen, 136
- Syntax
 - benutzerdefinierte Funktion, 167
 - der Namen von Variablen, 115
 - Objekte mit Einheiten, 202
 - Syntax eines Integrals, 462
- Systemflags, 241
- System-Flags, 555
 - Vollständige Liste der, 757

T

- Tage, Zahl der Tage zwischen zwei Daten, 492
- Tag/Monat/Jahr-Anzeige, 477
- Taktsteuerung beim Empfang
 - Einstellen, 671
- Taktsteuerung beim Senden
 - Einstellen, 671
 - während des Drucks, 660
- Taktsteuerung XON/XOFF, 671
- Taktsteuerung(Senden/Empfang),
 - Einstellen, 671
- Tangens, 150
- Tastatur
 - Akzentbuchstaben eintippen, 57
 - Aufbau, 26
 - Begrenzungszeichen eintippen, 58
 - benutzerdefinierte
 - Tastenbelegungen löschen, 238
 - “blaue” Tasten, 52
 - Buchstaben eingeben, 56
 - Datum eintippen, 476
 - ein Programm eingeben, 506
 - Eingabe von Sonderzeichen, 57
 - Funktionen, 144 – 145
 - griechische Buchstaben, 52
 - Großbuchstaben, 52, 56
 - Kleinbuchstaben, 52, 56
 - “orangefarbene” Tasten, 52
 - sechs Ebenen, 26, 52
 - Sonderzeichen, 52
 - speichert bis zu 15
 - Tastendrucke, 51
 - statistische Daten eintippen, 399
 - Uhrzeit eintippen, 477
 - umbelegen, 234
 - Umschalttasten, 55
 - Vektoren eintippen, 186
 - Verwenden der Rückschritttaste
 - bei Fehleingabe, 49
 - Zahlen eintippen, 49
 - Ziffernblock, 52
- Tastenadresse, ausgeben, 584
- Tastenbelegungen,
 - Benutzertastatur, 235
- Tastenfolgen in Warteschlange, 51
- Taylorpolynome
 - Berechnen des Taylorpolynoms
 - eines algebraischen Ausdrucks, 460
 - Näherung für den Integranden, 466
 - Verlegen des
 - Auswertungspunktes, 461
- Teil einer Liste oder Zeichenkette, 95
- Teilausdrücke
 - Auswahlumgebung, 432
 - Definition, 262, 428

- EquationWriter, 247
- Teilausdrücke im
 - EquationWriter bearbeiten, 263
- Teilausdrücke im
 - EquationWriter ersetzen, 266
- Temperatur, Auswirkung auf den Taschenrechner, 717
- Temperaturumrechnung, 213
- temporäre Variable, 112
- Temporäres Menü, Verwendung in Dialogprogrammen, 583
- Terme neu ordnen, Umformungen mit RULES, 431
- Termine
 - Akustisches Signal abschalten, 483
 - ausgeführte Aktion, 479
 - Befehle, 486
 - Bestätigen von Terminen, 482
 - Einstellen von Terminen, 479
 - neu ansetzen, 483
 - Sichern von Terminen, 483
 - Steuerungstermine, 484
 - Terminerinnerungen, 480
 - überfällige Termine, 483
 - Überprüfen und Bearbeiten von Terminen, 485
 - unbestätigte Termine, 483
 - Verwendung von Terminen in Programmen, 489
 - Wiederholen von Terminen, 479, 481
 - zu wiederholende Termine in zu kurzen Abständen, 484
- Terminerinnerungen
 - Bestätigen von Terminen, 482
 - unbestätigte Termine, 483
- Terminkatalog, 485
 - Operationen im Terminkatalog, 486

- Testen
 - Flags, 556
 - Funktion der Tastatur, 725
 - Funktion des Taschenrechners, 722
 - Selbsttest, 724
- Testen des Port-RAM, 726
- Testen des seriellen Anschlusses, 729
- Teststatistik, 415
- Text, in die Befehlszeile eingeben und bearbeiten, 48
- TICKS, 494
- Tiefe des Stacks, bestimmen, 83
- TIME-ADJST-Menü, 478
- TIME-ALRM-Menü, 479
- TIME-Menü, 475
- TIME-RPT-Menü, 481
- TIME-SET-Menü, 476
- transponierte Matrix, 388
- Trennen, von Objekten, 95
- Trennen der Variablen nach Objekttyp, 103
- trigonometrische Funktionen, 150
 - trigonometr. Funktionen mit ALGEBRA erweitern, 443
- trigonometrische Operationen, Objekte mit Einheiten, 219
- TRUTH-Plottyp, 353, 359
- Typ, Objekttyp-Nummer ausgeben, 532

U

- überbestimmte Systeme, 392
- überfällige Termine, 483
- übergeordnetes Verzeichnis, 127
- überlaufendes Sortieren, 608
- überprüfen, Flags, 241
- Übersetzungsmodus, während des Drucks, 660
- Übertragung, Indikator für, 51

- Übertragungsgeschwindigkeit,
 - Einstellung, 669
- Übertragungsrate, während des Drucks, 660
- Uhr
 - Befehle, 476
 - Ermittlung der Ausführungszeit, 598
 - Korrigieren der Uhrzeit, 478
- Uhrzeit
 - AM/PM-Anzeige, 477
 - Anzeige ändern, 477
 - Befehle, 476
 - einstellen, 477
 - Korrigieren der Uhrzeit, 478
 - vierundzwanzig-Stunden-Anzeige, 477
 - zwölf-Stunden-Anzeige, 477
- Umbelegen von Tasten, 235
- Umdefinieren der Tastatur, 234
- Umgebung, der Interaktive Stack, 75
- Umgebungen
 - Auswahl, 262
 - Auswahlumgebung, 431
 - Gleichungskatalog, 280
 - Grafikumgebung, 308, 324
 - Statistik-Katalog, 400
 - Terminkatalog, 485
- umgeschaltet links
 - Indikator für, 51, 55
- umgeschaltet rechts
 - Indikator für, 51, 55
- Umkehrung, einer Variablen, 123
- umrechnen, Einheiten, 203
- Umschalttaste, 26
 - links, 26
 - rechts, 26
- Umschalttasten, 55
 - Benutzermenüs, 233
- umsortieren
 - Gleichungskatalog, 281
 - Statistik-Katalog, 401
- umwandeln
 - binär in reell, 227
 - Datum in Zahl umwandeln, 491
 - Datum in Zeichenkette umwandeln, 491
 - Einheiten, 211
 - Grad in Radiant, 152
 - HMS in Zahl umwandeln, 493
 - komplex in reell, 95, 179
 - komplexes Feld in reelles Feld, 386
 - Objekte in Zeichenketten, 95
 - Objekte mit Einheiten, 209, 210
 - Pixelkoordinaten in
 - benutzerdefinierte Koordinaten umwandeln, 350
 - Radiant in Grad, 152
 - reell in binär, 227
 - reell in komplex, 95, 179
 - reelle Zahlen in Brüche, 146
 - reelles Feld in komplexes Feld, 386
 - Temperaturumrechnung, 213
 - von Zahlen zu Zeichen, 95
 - Zahl in Datum umwandeln, 491
 - Zahl in HMS umwandeln, 493
 - zusammengesetzte Einheiten in SI-Einheiten, 212
- unabhängige Stichproben, 404
- unabhängige Variable
 - für Plotten angeben, 317
 - Plotbereich, 345
 - Statistik, 406
- unabhängiger Speicher, 697
- unbenutzter Speicher (freier Speicher), 107
- unbestimmte Schleifen, 549
- UND, bei Tests, 531
- Units, 24
- UNITS, 200–223
 - Arithmetik für Objekte mit Einheiten, 216

- arithmetische Operationen, 220
 - Ausdrücke zerlegen, 215
 - benutzerdefinierte Einheiten, 222
 - eingebaute Einheiten, 209
 - Einheitenumrechnung, 203
 - Gleichung für ideale Gase, 200
 - Internationales Einheitensystem (SI), 202
 - Kehrwert eines Objektes mit Einheiten, 218
 - Namen von Einheiten auflisten, 206
 - Objekte mit Einheiten erzeugen, 203
 - Objekte mit Einheiten im EquationWriter erzeugen, 220
 - Objekte mit Einheiten im Menü CST umwandeln, 211
 - Objekte mit Einheiten in algebraischen Objekten, 207
 - Objekte mit Einheiten in der Befehlszeile erzeugen, 205
 - Objekte mit Einheiten potenzieren, 218
 - Objekte mit Einheiten umwandeln, 209, 210
 - Objekte mit Einheiten vergleichen, 219
 - Objekte mit Einheiten zerlegen, 223
 - Objekte mit Einheiten zusammensetzen, 223
 - photometrische Einheiten, 215
 - Priorität von Funktionen, 207
 - Prozentberechnungen, 219
 - Temperaturumrechnung, 213
 - trigonometrische Operationen, 219
 - Umwandlung in SI-Basiseinheiten, 212
 - UNITS-Katalogmenü, 202, 203
 - Vorsätze, 208
 - Vorsätze für benutzerdefinierte Einheiten, 223
 - Winkleinheiten, 214
 - UNITS-Anwendung, Objekte mit Einheiten eingeben und bearbeiten, 203
 - UNITS-Befehlsmenü, 202
 - UNITS-Katalogmenü, 202, 203, 209
 - unterbestimmte Systeme, 392
 - unterbrechen
 - Lösungsalgorithmus, 299
 - von Programmen mit der ATTN-Taste, 58
 - Unterbrechung
 - Einstellung, 685
 - Programme mit dem Befehl HALT, 521
 - Unterbrechungsindikator, 51
 - Unterprogramme, 518
 - Einzelschrittausführung, 524
 - Unterverzeichnisse, 126
 - Anlegen von Unterverzeichnissen, 128
 - bearbeiten, 132
- ## V
- Variable
 - Anschauen und Bearbeiten, 72
 - Anzeigen der Objekttypen-Nummer des gespeicherten Objektes, 102
 - belegter Speicherplatz, 107
 - Namen, 89
 - Trennen der Variablen nach Objekttyp, 103
 - Variablen, 112–125
 - alle V. in einem Verzeichnis löschen, 123
 - Arithmetik mit Variablen, 123

- auswerten, die Programme
 - enthalten, 117
 - Benutzermenü, 231
 - das Menü VAR neu ordnen, 121
 - definieren, 115
 - den Namen auswerten, 116
 - die Namen enthalten, 118
 - die Verzeichnisse enthalten, 118
 - doppelte Namen, 130
 - Druck, 656
 - erzeugen, 113, 114
 - globale Variablen, 112
 - in Variablen gespeicherte
 - Variablen, 427
 - Inhalt ändern, 118
 - Inhalt aufrufen, 118
 - Katalog REVIEW, 120
 - lokale Variablen, 112
 - löschen, 122
 - Menüfelder, 113, 115
 - Namen, 115
 - Namen kopieren, 130
 - Namen mit und ohne
 - Anführungszeichen, 119
 - normale Variablen, 112
 - reservierte, 115
 - sind benannte Speicherplätze,
 - 112
 - sofort ausführen, 119
 - speichern, 114
 - temporäre Variablen, 112
 - V. in anderen Verzeichnissen,
 - 130
 - versehentlich gelöscht, 123
 - Verwenden des Inhalts, 116
 - Verwenden lokaler V., 113
 - Verzeichnisobjekt als Inhalt, 132
 - Verzeichnisse speichern, 126
 - während der Auswertung nach
 - dem Namen suchen, 130
- VAR-Menü, 113, 120, 126
 neu ordnen, 121
- VECTR-Menü, 152, 185, 198
- Vektoren, 88, 183–199
- Anzeige, 184
 - Anzeigemodi, 378
 - Arithmetik, 382
 - Arithmetik mit Matrizen, 384
 - Befehle, 198
 - Berechnungen, 190
 - Betrag, 190, 382
 - das *n*te Element auslesen, 95
 - einen Vektor durch eine Matrix
 - dividieren, 385
 - Einheitsvektor, 191
 - eintippen, 186
 - Felder, 373
 - interne Darstellung, 185
 - komplexe Vektoren, 386
 - Kreuzprodukt, 190, 382
 - PUT ersetzt das *n*te Element,
 - 95
 - Skalarprodukt, 190, 382
 - Vergleich mit komplexen
 - Zahlen, 179, 180, 199
 - zerlegen, 187, 198
 - zusammensetzen, 187, 198
- Verarbeiten, von lokalen
 - Variablen, 616
- Verbinden von Gleichungen im
 - Gleichungskatalog, 294
- Verbindung, Algebra, 438
- verbundene Stichproben, Statistik,
 - 406
- verdeckte Variablen, Anzeigen
 - verdeckter Variablen, 427
- Vereinfachen algebraischer
 - Objekte, 137
- verfügbarer Speicher, Zahl
 - unbenutzter Bytes, 107
- Vergleichsfunktionen, 529
 - in algebraischen Ausdrücken,
 - 530
- Verkürzen von Zahlen, 159
- verschachteln, benutzerdefinierte
 - Funktionen, 165

verschachtelte Klammern,
algebraische Objekte, 138
verschachtelte Schleifen, 608
Verschieben von Termen, Algebra,
436
Vertauschung, Algebra, 438
Verteilung, Algebra, 438
Verzeichnispfad, 127
wird in den Statuszeilen
angezeigt, 50
Verzeichnisse
aktuelles Verzeichnis, 127
alle Variablen eines bestimmten
Typs in einem V.
bestimmen, 103
anlegen, 132
Anlegen von Verzeichnissen,
128
aufrufen, 132
Begriffe, 126
HOME-Verzeichnis, 126
in die nächsthöhere Ebene
wechseln, 131
in Variablen, 118
löschen, 131
neue Variablen werden ins
aktuelle V. aufgenommen,
129
Objekttypen-Nummer, 102
übergeordnetes Verzeichnis, 127
V. während des Auswertens
nach dem Variablennamen
absuchen, 130
Verzeichnispfad, 127
vierundzwanzig-Stunden-Anzeige
der Uhrzeit, 477
vom PC zum HP 48
Eingabe/Ausgabe, 674
Kabelverbindung, 674
vorherige Ergebnisse in
Kettenrechnungen
verwenden, 67
Vorhersagewert, 406

Vorsätze für benutzerdefinierte
Einheiten, 223
vorstellen, Uhr vorstellen, 478
Vorwärtszählen, der
Programmschleifenzähler,
553
Vorzeichen
Änderung des Vorzeichens
einer Zahl, 49
bestimmen, 159
Objekte mit Einheiten, 220
Vorzeichenwechsel, 144

W

“Wahr”, Ergebnis eines Tests, 528
Wahrscheinlichkeit, 158
WAIT
Unterbrechung eines
Programmablaufs, 578
Verwenden des Argumentes 0,
584
Warteschlange, Tastenfolgen in, 51
Wartung, 732
Überprüfung der
Taschenrechnerfunktion, 722
wechseln ins Stammverzeichnis
oder das Verzeichnis
HOME, 131
Wert einer symbolischen
Konstanten, 155
WHILE...REPEAT...END, 551
Winkel, komplexe Zahlen, 170
Winkleinheiten, 214
Winkelmodi, 378
Winkelmodus, 149, 184
wählen, 149
Winkelumrechnungsfunktionen,
152
wissenschaftliche
Zahlendarstellung, Eingabe
von Exponent und Mantissee,
49

wobei-Funktion, 450
EquationWriter, 255
Wortlänge, Binärzahlen, 224
Wurzel, Ermitteln der Quadrat-
oder x ten Wurzel einer Zahl,
144

X

x -Achse, Anzeigebereich festlegen,
318
XLIB-Namen, Objekttypen-
Nummer, 102
XON/XOFF-Quittungsbetrieb,
während des Drucks, 660
XOR
bei Tests, 531
Binärzahlen, 227
 χ^2 -Test, 415
 x -te Wurzel, EquationWriter, 251

Y

y -Achse, Anzeigebereich festlegen,
318
Zahlen

Z

Anzeigemodi, 61
EquationWriter, 248
in die Befehlszeile eintippen, 49
in Zeichen umwandeln, 95
interne Darstellung, 61
Zahl in Datum umwandeln, 491
Zahlenpaare, können als
komplexe Zahlen dargestellt
werden, 86
Zahlensystem wählen, 87
Zähler, EquationWriter, 247
Zehnerlogarithmus, 147
Zehnerpotenzen, EquationWriter,
252

Zeichen

Akzente eingeben, 57
Bestimmen des numerischen
Werts, 95
eine Untermenge bilden, 95
Eingeben von Sonderzeichen, 52
Zahlen in Zeichen umwandeln,
95
Zeichen umsetzen,
Eingabe/Ausgabe, 679
Zeichencodes, 752–754
Zeichenketten
abgezählte Zeichenketten, 91
aus einem Objekt, 95
Druck, 654, 656
einen Teil ersetzen, 95
in Objekt umwandeln, 95
Inhalt auswerten, 95
Objekttypen-Nummer, 102
Position innerhalb einer
anderen Zeichenkette, 95
Zahl der Zeichen (Größe), 95
zusammenfügen, 95
Zeichensätze
Druck des HP 48-
Zeichensatzes, 658
Druck mit dem Infrarot-
Drucker, 660
für Infrarotdrucker umsetzen,
653
Umsetzen während
Eingabe/Ausgabe, 679
zeichnen, 372
Achsbeschriftungen und
Koordinatensprung, 346
Auflösung, 346
benutzerdefinierte und
Pixelkoordinaten, 349
Größe von *PICT* ändern, 351
Koordinatensysteme, 349
Parameter in *PPAR* speichern,
347

- Plotbereich der abhängigen und der unabhängigen Variablen, 345
- Plotparameter zurücksetzen und *PICT* löschen, 348
- Statistik, 409
 - unabhängige Stichprobe, 409
 - verbundene Stichprobe, 409
 - Verfeinerungsmöglichkeiten, 343
- Zeilenabschluß, während des Drucks, 660
- Zeilenlänge, während des Drucks, 660
- Zeilen-Norm, berechnen, 388
- Zeilenvorschub, Speicherinhalt des Druckerpuffers ausgeben, 653
- Zeit, erforderlich zur Ausführung eines Programms, 597
- Zeitarithmetik, 492
- Zeitlich begrenzt gültige Variablen, verwendet in Programmen, 510
- Zentrieren einer Zeichnung, 318
- zerlegen
 - komplexe Zahlen, 173, 179
 - Vektoren, 187, 198
 - von Objekten mit Einheiten, 223
- Zerlegen von Ausdrücken für Einheiten, 215
- Ziffernblock, der Tastatur, 52
- zoomen, ein Rechteck vergrößern, 330
- ZOOM-Menü, 329
- Zoomoperationen, 328
- zu geringe Batteriespannung, Batterien austauschen, 717
- zu wenig Speicherplatz, 109, 110
- Zufallszahlen, auswählen, 158
- zukünftiges Datum, Berechnen eines Datums, 491
- zurücksetzen
 - Plotparameter, 315, 348
- zurückstellen, Uhr zurückstellen, 478
- Zusammenfassen von Termen, 428
 - Algebra, 436
- zusammenfügen, komplexe Zahlen, 179
- Zusammengefaßter Speicher, 697
- zusammensetzen, komplexe Zahlen, 173
- Zusammensetzen, Vektoren, 187
- zusammensetzen
 - Vektoren, 198
 - von Objekten mit Einheiten, 223
- Zustandsmarke für den Zeilenvorschub, 659
- Zustandsmarken
 - E/A-Einrichtung, 660
 - Zeilenvorschub, 659
- zweidimensionale Punkte, können als komplexe Zahlen dargestellt werden, 86
- zweidimensionale Vektoren, 88
- Zweizeiliger Druck, 657
- zwölf-Stunden-Anzeige der Uhrzeit, 477
- Zylinderkoordinaten-Modus, Indikator für, 51
- Zylindermodus, 184
 - Indikator, 184

Unterstützung durch Hewlett-Packard

Informationen über die Verwendung des Taschenrechners.

Wenn Sie Fragen über die Verwendung des Taschenrechners haben, sollten Sie zunächst das Inhaltsverzeichnis, das Register und den Abschnitt "Antworten auf häufige Fragen" in Anhang A zu Rate ziehen. Falls das Handbuch keine befriedigende Antwort enthält, können Sie sich mit dem Reparaturzentrum in Frankfurt in Verbindung setzen:

Reparaturzentrum Frankfurt
Berner Straße 117
6000 Frankfurt 56
Tel. (069) 5 00 06-0

Kundendienst. Wenn Sie den Eindruck haben, daß Ihr Taschenrechner nicht ordnungsgemäß funktioniert, sollten Sie die Hinweise zur Fehlersuche und die Informationen zum Kundendienst in Anhang A konsultieren. Wenn Sie sich in den Vereinigten Staaten befinden und Ihr Taschenrechner wartungsbedürftig sein, dann können Sie ihn per Post an das Corvallis Service Center schicken:

Hewlett-Packard
Corvallis Service Center
1030 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.
(503) 757-2002

Inhaltsverzeichnis

Teil 4: Programmierung

Seite	504	25: Grundlagen der Programmierung
	526	26: Tests und Bedingte Strukturen
	540	27: Aufbau von Programmschleifen
	555	28: Flags
	560	29: Dialogprogramme
	586	30: Fehlerbehandlung
	593	31: eitere Programmierbeispiele

Part 5: Drucken, Datenübertragung und Einsteckmodule

Seite	652	32: Drucken
	663	33: Datenübertragung vom und zum HP 48
	688	34: Verwendung von Einsteckkarten und Bibliotheken

Anhänge und Verzeichnisse

Seite	712	A: Unterstützung, Batterien und Wartung
	735	B: Meldungen
	752	C: Zeichencodes des HP 48
	755	D: Menünummern
	757	E: Zusammenstellung der System-Flags des HP 48
	766	Verzeichnis der Operationen
	881	Index



Bestellnummer

00048-90003

00048-90008 Deutsch

Printed in West Germany 12/90