

LOGO- HANDBUCH

© 1984 S.O.L.I. (Paris)
Systèmes d'Ordinateur Logo International
33, rue de Poissy
F-75005 Paris

© 1984 CEDIC
32, boulevard Saint-Germain
F-75005 Paris

© 1986 Thomson Micro GmbH
Dreieichstraße 10
D-6082 Mörfelden-Walldorf

Alle Rechte vorbehalten. Kein Teil dieses Buches darf ohne ausdrückliche Genehmigung des Verlags in irgendeiner Form reproduziert oder übermittelt werden, weder in mechanischer noch in elektronischer Form, inkl. Fotokopie.

Satz und Druck:
Munkelt Druck GmbH, 6100 Darmstadt, Deutschland

INHALTSVERZEICHNIS

Vorwort	7
Einschalten des Systems	9
Die Tastatur	11
Teil I	
Die Struktur der Programmiersprache Logo	
Kapitel 1: Der Aufbau	13
1.1 Definitionen	13
1.2 Beispiele	14
Kapitel 2: Grundprozeduren und definierte Prozeduren	16
2.1 Definieren und Ausführen einer Prozedur	16
2.2 Prozeduren mit Parametern	18
Kapitel 3: Befehle und Operationen	20
Kapitel 4: Die Namen in den Prozeduren	22
Kapitel 5: Die Logo-Programmzeilen	24
Kapitel 6: Sonderzeichen	27
Teil II	
Die Grundwörter der Programmiersprache Logo	
Kapitel 1: Die Welt des Igels	31
1.1 Die Bewegungen	32
1.2 Das Igelfeld	36
1.3 Der Zeichenstift	44
Kapitel 2: Die Welt der Wörter und der Listen	48
2.1 Prüfen	49
2.2 Abbauen	53
2.3 Aufbauen	55
Kapitel 3: Die logischen Grundoperationen	58
Kapitel 4: Die Prozeduren	61
4.1 Definieren	61
4.2 Prüfen	63
	5

Kapitel 5: Die Grundwörter zur Programmsteuerung	65
Kapitel 6: Die Namen	71
Kapitel 7: Arithmetik	73
7.1 Die arithmetischen Operationen	75
7.2 Die Vergleiche	78
7.3 Weitere arithmetische Funktionen	79
Kapitel 8: Die Verwaltung des Arbeitsspeichers	80
8.1 Prüfen	81
8.2 Ändern	83
Kapitel 9: Die Verbindung mit der Außenwelt	84
9.1 Die Tastatur	85
9.2 Der Bildschirm	86
9.3 Der Light-Pen	90
9.4 Die Joysticks	91
9.5 Die Musik	92
9.6 Die anderen Peripheriegeräte	94
Kapitel 10: Die Dateien	96
Kapitel 11: Die Welt des Editors	100
11.1 Aufrufen und Verlassen des Editors	100
11.2 Die Arbeit mit dem Editor	102
Kapitel 12: Grundwörter für den erfahrenen Benutzer	105
Anhang A: Die Logo-Meldungen	106
Anhang B: Die ASCII-Codes und die Benutzung von Sonderzeichen	109
Anhang C: Automatisches Laden von Prozeduren	113
Anhang D: Endliche und unendliche Prozeduren	114
Anhang E: Verzeichnis der Befehle und Operationen	115
Anhang F: Alphabetisches Grundwörterverzeichnis	121
Anhang G: Verzeichnis der definierten Prozeduren	124

Vorwort

Dieses Handbuch soll Sie bei der Logo-Programmierung unterstützen und setzt voraus, daß Sie bereits über einige Logo-Kenntnisse verfügen. Sollte das nicht der Fall sein, empfehlen wir Ihnen zunächst die Lektüre des Logo-Einführungsbuches.

Der erste Teil dieses Handbuchs behandelt in kurzer Form die Struktur der Programmiersprache Logo sowie die „Grammatik“. Im zweiten Teil werden alle Grundwörter („eingebaute“ Befehle und Operationen) vorgestellt und anhand von Beispielen erläutert.

Im Anhang finden Sie alle Logo-Meldungen, die ASCII-Code-Tabelle, ein nach Kapiteln geordnetes und ein alphabetisches Grundwortverzeichnis, sowie weitere Informationen.

Einschalten des Systems

Damit Sie Logo auf Ihrem System benutzen können, sind zunächst folgende Schritte erforderlich:

1. Legen Sie das Logo-Programmmodul in das Programm-Modul-Fach ein.
2. Schalten Sie das Fernsehgerät ein, dann die vorhandenen Peripheriegeräte, und zuletzt den Computer.
3. Wenn Sie einen MO5 E besitzen, steht Logo jetzt zu Ihrer Verfügung. Beim TO7-70 drücken Sie bitte die Taste mit der Ziffer 1 (für Logo). Daraufhin erscheint folgende Meldung auf dem Bildschirm:

```
LOGO 1.0 (c) SOLI 1984
?_
```

4. Wenn Sie die Diskettenstation verwenden wollen, müssen Sie wie folgt vorgehen:

MO5 E

Nach Schritt 3 die LOGO-DOS-Diskette einlegen und .EFT tippen (.EFT reinitialisiert LOGO)

TO7-70

Bevor Sie im Schritt 3 die 1 anwählen, die LOGO-DOS-Diskette einlegen.

5. Wenn Sie LOGO starten wollen und es automatisch ein Programm laden soll, schauen Sie im Anhang C nach.

Schalten Sie, falls die System-Meldung nicht erscheint, noch einmal alle Geräte aus (zuerst den Computer) und überprüfen Sie, ob das Programmmodul richtig eingelegt, das Programm-Modul-Fach geschlossen und alle Verbindungen hergestellt sind. Wiederholen Sie anschließend den Einschaltvorgang wie beschrieben. Wenn die Logo-Systemmeldung dann noch immer nicht erscheint, kann ein Gerätefehler vorliegen. Fragen Sie in diesem Fall Ihren Fachhändler.

Die Logo-Systemmeldung zeigt an, daß Logo in den Arbeitsspeicher geladen wurde. Das Fragezeichen zu Beginn der Folgezeile ist die Logo-Eingabeaufforderung. Wenn Sie erscheint, wartet Logo auf Ihre Befehle. Die blinkende Unterstreichung hinter dem Fragezeichen ist der sogenannte Cursor. Er kennzeichnet die Position, an der Logo Ihre Eingabe erwartet.

Die Tastatur

Die Tastatur enthält alle alphanumerischen Zeichen (Buchstaben, Ziffern und Satzzeichen), die auch auf einer normalen Schreibmaschine vorhanden sind. Darüberhinaus finden Sie einige speziell für Ihren Computer bestimmte Zeichen und Tasten.

[ENTER] beendet eine Eingabe.

[] Leertaste. Schreibt ein Leerzeichen und versetzt den Cursor um ein Zeichen nach rechts.

[←] löscht das Zeichen links vom Cursor und setzt den Cursor unter das gelöschte Zeichen.

[CNT] bzw. [CTRL] Control-Taste. Durch Drücken der CNT- bzw. der CTRL-Taste und gleichzeitigem Drücken einer anderen Taste können bestimmte Systemfunktionen ausgeführt oder Sonderzeichen dargestellt werden. Beispiel:

[CNT] bzw. [CTRL][A] bewirkt die Ausgabe des Zeichens [A]
[CNT] bzw. [CTRL][Z] bewirkt die Ausgabe des Zeichens [Z]

[STOP] unterbricht die Programmausführung. Um das Programm fortzusetzen, genügt es, eine beliebige Taste zu drücken.

Im Editiermodus:

[←] verschiebt den Cursor nach links.

[→] verschiebt den Cursor nach rechts.

[↑] verschiebt den Cursor nach oben.

[↓] verschiebt den Cursor nach unten.

Im Editiermodus:

[INS] In Verbindung mit einer anderen Taste können bestimmte Editor-Funktionen ausgeführt werden.

Im Editiermodus:

[DEL] löscht das Zeichen über dem Cursor.

[▲] Durch gleichzeitiges Drücken dieser Taste und einer Taste mit einem grünen Zeichen, wird das grüne Zeichen ausgegeben.

Hinweis:

Beim MO5 E entspricht die Taste [▲] der Taste [SHIFT].

Nach gleichzeitigem Drücken der Tasten [▲] und [] erscheinen alle Buchstaben in Kleinschrift. Durch erneutes gleichzeitiges Drücken beider Tasten wird wieder auf Großschrift umgeschaltet.

[ACC] Mit Hilfe dieser Taste lassen sich Akzentzeichen setzen und Umlaute darstellen (nur in Kleinschrift).

Dazu müssen Sie folgende Tasten drücken: [ACC], dann [▲] und den gewünschten Akzent und schließlich den Buchstaben.

Für die deutschen Umlaute brauchen Sie nur folgende Tasten zu drücken:

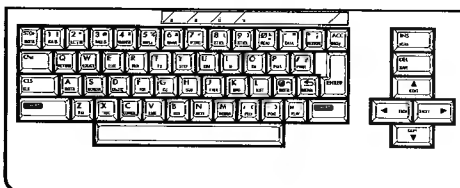
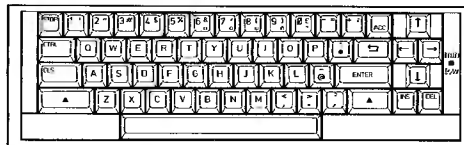
[ACC] und 6 für ä

[ACC] und 7 für ö

[ACC] und 8 für ü

Alle Logo-Anweisungen müssen in Großbuchstaben geschrieben werden. Nur Texte, die keine Befehle oder Operationen sind, dürfen Kleinbuchstaben enthalten.

Die Tastatur des TO7-70



Die Tastatur des MO5 E

TEIL I

Die Struktur der Programmiersprache Logo

Kapitel I/1 Der Aufbau

1.1 Definitionen:

a) Die Logo-Terme

Logo arbeitet mit Wörtern und Listen. Jede zusammenhängende Zeichenfolge bildet ein Wort. Mehrere Wörter ergeben eine Liste. Wörter und Listen werden unter dem Begriff „Terme“ zusammengefaßt. Ein Logo-Term ist also entweder ein Wort oder eine Liste.

Die Wörter

Ein Wort ist eine zusammenhängende Zeichenfolge, die nicht durch ein Leerzeichen getrennt ist. Bei den Zeichen kann es sich um Ziffern, Buchstaben oder andere Zeichen der Tastatur handeln. Ein Wort ist für Logo zunächst nur eine sinnlose Zeichenfolge. Es ist Ihre Aufgabe, dem Wort entsprechend den Logo-Regeln eine Bedeutung zuzuweisen!

Die Listen

Eine Liste ist eine Folge von Listenelementen. Die Elemente einer Liste können Wörter oder selbst wieder Listen sein.

b) Namen und Objekte

Ein Name ist ein Wort, das einen Logo-Term bezeichnet. Terme, denen ein Name zugeordnet ist, heißen Objekte.

c) Die Prozeduren

Eine Prozedur ist eine Folge von Aktionen (oder nur eine Aktion), die von Logo ausgeführt werden kann. Wie einem Term wird dieser Aktionsfolge ein Name (Wort) zugeordnet. Da Logo nach Aufruf dieses Namens die zugehörigen Aktionen ausführt, bezeichnet man den Namen selbst als Prozedur.

1.2 Beispiele

- *Wörter:*

```
?DZ "AZ23Ka  
AZ23Ka
```

Das Wort DZ (Drucke Zeile) ist eine Prozedur, die mit dem Parameter "AZ23Ka arbeitet. Zahlen sind besondere Worte:

```
?DZ 2578  
2578
```

- *Listen:*

```
?DZ [ A B C ]  
A B C
```

```
?DZ [HALLO [VW 30] D2 R2]  
HALLO [VW 30] D2 R2
```

Ein Wort wird zu einem Namen, sobald Sie mit dem Wort einem Logo-Term bezeichnen:

```
?SEI "GRUSS "HALLO
```

Das Wort SEI ist eine Prozedur; die Wörter HALLO und GRUSS die zugehörigen Parameter. Die Prozedur SEI dient hier dazu, dem Wort HALLO den Namen GRUSS zu geben. Damit ist GRUSS nicht mehr nur eine beliebige Zeichenfolge, sondern der Name des Logo-Terms HALLO, und HALLO ist der Parameter von GRUSS.

– *Prozeduren:*

```
?PR GRUSS  
>DZ "HALLO  
>ENDE  
SIE DEFINIEREN GERADE GRUSS
```

Die Prozedur PR (Prozedur) hat der Aktion DZ "HALLO (DZ=Drucke Zeile) den Namen GRUSS zugewiesen. GRUSS ist nun selbst eine Prozedur.

```
?GRUSS  
HALLO
```

Die Programmiersprache Logo dient dazu, Programme zu erstellen. Ein Logo-Programm besteht aus Modulen, die auf unterschiedliche Weise kombiniert werden können, wobei natürlich bestimmte Regeln einzuhalten sind. Dieses wenigen und einfachen Regeln bilden die „Grammatik“ der Programmiersprache.

Kapitel I/2

Grundprozeduren und definierte Prozeduren

Bestimmte Logo-Prozeduren stehen sofort nach dem Einschalten zur Verfügung. Es handelt sich dabei um die „eingebauten“ Grundprozeduren, die auch als „Grundwörter“ bezeichnet werden und zum standardmäßigen Befehlsumfang von Logo gehören.

Geben Sie z.B. ein:

```
?LB (Lösche Bildschirm)
```

Die Spuren des Igels verschwinden vom Bildschirm, obwohl Sie die Prozedur LB zuvor nicht definiert haben. Sie gehört zu den Grundprozeduren, und Logo kann sie deshalb sofort ausführen.

Neben den Grundprozeduren gibt es diejenigen Prozeduren, die Sie selbst mit Hilfe des Grundwortes PR definiert haben. Wir werden Sie in den folgenden Kapitel einfach nur als Prozeduren bezeichnen.

2.1 Definieren und Ausführen einer Prozedur

Beispiel für die Definition einer Prozedur:

```
PR HALLO  
DZ "HALLO  
ENDE
```

Die erste Zeile ist die Kopf- oder Titelzeile der Prozedur. Sie beginnt immer mit dem Grundwort PR. Anschließend folgt der Name der Prozedur. Er darf niemals mit einem Anführungszeichen (") oder einem Doppelpunkt (;) beginnen.

Die letzte Zeile einer Prozedur besteht immer aus dem Grundwort ENDE.

Um eine Prozedur auszuführen, müssen Sie sie „aufrufen“.

Die mittlere Zeile der Prozedur HALLO ist eine Aufforderung an Logo, das Wort HALLO zu schreiben (DZ = drucke Zeile). Wir können auch sagen: Die zweite Zeile ruft die Grundprozedur DZ auf, um das Wort HALLO auszugeben.

Es gibt aber noch eine weitere Möglichkeit, um eine Prozedur aufzurufen. Es genügt nämlich, im Direktmodus einfach den Namen der Prozedur einzugeben (der Direktmodus wird angezeigt durch das Fragezeichen am Anfang der Zeile). Sie haben diese Möglichkeit bereits mit dem Befehl LB ausprobiert.

Dazu ein weiteres Beispiel:

```
?DZ "HALLO  
HALLO  
  
?HALLO  
HALLO
```

Wenn Logo einen eingegebenen Prozedurnamen nicht finden kann, wird eine Fehlermeldung ausgegeben. Wenn Sie z.B. die Prozedur VIERECK noch nicht definiert oder die definierte Prozedur nicht in den Arbeitsspeicher geladen haben, erhalten Sie nach dem Aufruf von VIERECK eine entsprechende Meldung:

```
?VIERECK  
PROZEDUR UNBEKANNT VIERECK
```

Bei der Definition einer Prozedur dürfen Sie bereits definierte Prozeduren aufrufen:

```
PR GRUSS  
HALLO  
DZ [UND GUTEN TAG]  
ENDE  
  
?GRUSS  
HALLO  
UND GUTEN TAG
```

2.2 Prozeduren mit Parametern

Ein Parameter ist eine Information, die beim Aufruf einer Prozedur angegeben werden muß, damit die Prozedur wie gewünscht arbeiten kann. Der Parameter darf ein Wort oder eine Liste sein.

Auch bestimmte Grundprozeduren arbeiten mit Parametern, z.B.:

```
?DZ "HALLO  
HALLO
```

Wenn der erforderliche Parameter nicht angegeben wird, gibt Logo eine Fehlermeldung aus:

```
?DZ  
FEHLENDE EINGABEN FUER DZ
```

Die Anzahl und Art der erforderlichen Parameter hängt von der jeweiligen Prozedur ab.

```
?VW 45
```

VW (vorwärts) ist eine Grundprozedur, die als einzigen Parameter eine Zahl erwartet.

```
?WH 4 [VW 50 RE 90]
```

WH (wiederhole) benötigt zwei Parameter: eine Zahl und eine Liste.

Wenn Sie eigene Prozeduren mit Parametern definieren, müssen die Namen der Parameter, jeder mit einem Doppelpunkt beginnend, im Prozedurtitel angegeben werden, und zwar im Anschluß an den Prozedurnamen. Als Parametername ist jeder beliebige Name erlaubt. Der aktuelle Parameterwert wird unmittelbar vor dem Aufruf der Prozedur zugewiesen. Beispiel:

```
PR GRUSS1 :VORNAME  
DZ "HALLO  
DZ :VORNAME  
DZ [WIE GEHT ES DIR?]  
ENDE
```

An der Titelzeile der Prozedur GRUSS1 erkennt Logo, daß die Prozedur einen Parameter mit dem Namen VORNAME verarbeitet. Im Prozedurrumpf erfolgt dreimal der Aufruf des Grundwortes DZ. Im zweiten Aufruf wird die Ausgabe des Parameters VORNAME gefordert. Soll nun die Prozedur GRUSS1 ausgeführt werden, so müssen Sie den Parameter beim Aufruf der Prozedur spezifizieren:

```
?GRUSS1 "PETER  
HALLO  
PETER  
WIE GEHT ES DIR?
```

Logo hat das Wort PETER dem Parameter VORNAME zugeordnet, d.h., es hat das in der Titelzeile angegebene Wort PETER zum „Inhalt“ des Parameters VORNAME gemacht. Somit entspricht der Aufruf DZ :VORNAME im Prozedurrumpf dem Aufruf DZ "PETER.

Kapitel I/3

Befehle und Operationen

Einige Prozeduren erzeugen nach ihrem Aufruf bestimmte Terme. Wenn diese Terme nicht als Parameter an eine andere Prozedur übergeben werden, gibt Logo eine Fehlermeldung aus. Beispiel:

```
?SUMME 31 28  
WAS SOLL GESCHEHEN MIT 59
```

```
?DZ SUMME 31 28  
59
```

Es gibt also zwei Arten von Prozeduren. Die einen erzeugen immer einen Term; wir nennen sie Operationen. Die anderen, die keine Terme erzeugen, wollen wir Befehle nennen. Diese Unterscheidung ist wichtig, und wir werden sie auch für die Grundwörter benutzen.

Der durch eine Operation erzeugte Term ist das „Ergebnis“ der Operation.

Nach unserer Definition sind also beispielsweise DZ und SEI Befehle, ZZ und SUMME Operationen.

```
?DZ ZZ 2
```

Logo wird eine positive ganze Zufallszahl kleiner als 2 anzeigen, also 0 oder 1. Es benutzt dazu das Ergebnis der Operation ZZ 2 als Parameter für die Prozedur DZ. Sobald das Ergebnis festliegt, wird es an DZ übergeben und DZ ausgeführt.

Wenn Sie versuchen, einen Befehl als Parameter zu benutzen, erhalten Sie eine entsprechende Fehlermeldung:

```
?DZ SEI "X 25  
FEHLENDE EINGABEN FUER DZ
```

SEI ist ein Befehl und kann nicht als Parameter benutzt werden, denn SEI führt zwar eine Aktion aus, erzeugt jedoch keinen Term, der an DZ übergeben werden kann. Der Prozedur DZ fehlt folglich der Parameter.

Bisher haben wir als Beispiele nur die eingebauten Prozeduren benutzt. Die Unterteilung in Operationen und Befehle gilt aber ebenso für definierte Prozeduren. So ist z.B. die Prozedur GRUSS1 ein Befehl, denn sie liefert kein Ergebnis.

Beispiel für eine Operation:

```
PR NULL-ODER-EINS
WENN GLEICH? ZZ 2 0 [RG "NULL"][RG "EINS!"]
ENDE
```

Diese Prozedur schreibt das Wort „NULL“, wenn ZZ 2 gleich 0 ist, und „EINS“, wenn ZZ 2 den Wert 1 ergibt. Es handelt sich also um eine definierte Operation.

```
?NULL-ODER-EINS
WAS SOLL GESCHEHEN MIT NULL
```

oder

```
WAS SOLL GESCHEHEN MIT EINS
```

Richtig ist:

```
?DZ NULL-ODER-EINS
NULL
```

oder

```
EINS
```

Um eigene Operationen zu definieren, müssen Sie immer das Grundwort RG (Rückgabe) benutzen. Weitere Einzelheiten dazu finden Sie im zweiten Teil dieses Handbuchs.

Kapitel I/4

Die Namen in den Prozeduren

Betrachten wir noch einmal die Prozedur GRUSS1 :VORNAME. Beim Aufruf dieser Prozedur wird dem Parameter VORNAME der Term PETER zugeordnet. Diese Zuordnung gilt, bis nach Ausführung der Prozedur das Fragezeichen wieder auftaucht. Solange ist VORNAME der Name des Objekts PETER. Wir können auch sagen: VORNAME hat den „Inhalt“ bzw. den „Wert“ PETER.

Nach Beendigung der Prozedur befindet sich der Parameter VORNAME wieder im gleichen Zustand wie vor dem Aufruf der Prozedur, d.h., er hat keinen „Inhalt“ mehr.

Beispiel:

```
PR HALLO :VORNAME
DZ :VORNAME
ENDE
```

Während der Ausführung der Anweisung HALLO "PETER ist VORNAME der Name des Terms PETER. Nach der Ausführung bezeichnet VORNAME keinen Term mehr.

Es beeinflusst die Ausführung von HALLO "PETER nicht, wenn vorher z.B. die Zuweisung SEI "VORNAME "HELGA erfolgt ist. Nach der Ausführung von HALLO "PETER hat VORNAME immer noch den zugewiesenen Inhalt HELGA.

```
PR HALLOHALLO :VORNAME
GRUSS2 :VORNAME
DZ [AUF WIEDERSEHEN] DZ :VORNAME
ENDE
```

```
PR GRUSS2 :X
DZ "HALLO DZ :X
ENDE
```

HALLOHALLO ruft GRUSS2 auf. Bei der Ausführung der Anweisung HALLOHALLO "PETER erhält X in dem Moment, in dem GRUSS2 aufgerufen wird, den Inhalt PETER. Nach Beendigung von GRUSS2, d.h., nachdem der Text HALLO PETER ausgegeben worden ist, hat X keinen Inhalt mehr, während VORNAME noch immer den Inhalt PETER hat.

Ein anderes Beispiel:

```
PR SCHREIBE-INHALT :X
DZ WERT :X
ENDE
?SEI "Y 6
```

Wir haben damit dem Wort Y den Inhalt 6 gegeben.

```
?SCHREIBE-INHALT "Y
6
```

Während der Ausführung der Prozedur erhält X den Inhalt des Wortes Y.

:X enthält Y und WERT :X ergibt 6, da 6 der Inhalt von Y ist.

```
?SEI "X 4
```

Wir haben X den Inhalt 4 gegeben.

```
?SCHREIBE-INHALT "X
X
```

Während der Ausführung erhält :X den Inhalt X. Nach der Ausführung hat X wieder den Inhalt, den es zu Beginn der Ausführung hatte:

```
?DZ WERT "X
4
```


Kapitel I/5

Die Logo-Programmzeilen

Eine Programmzeile besteht in Logo aus einer Folge von Anweisungen (oder auch nur einer Anweisung), die durch Drücken der ENTER-Taste abgeschlossen wird. Im Direktmodus können maximal 248 Zeichen pro Zeile eingegeben werden. Bei einer Eingabe im Editiermodus (s. Kapitel II/11) ist die Zeilenlänge unbegrenzt. Eine Programmzeile kann sich also über mehrere Bildschirmzeilen erstrecken. Beispiel:

```
?SEI "NAMENLISTE [ANNE BERNHARD HELENE  
ERICH FRANZ MARIANNE KLAUS]
```

Da die Programmzeilen sehr komplex sein können, wollen wir versuchen, sie zu analysieren und bestimmte Kriterien herausfinden.

1. Sobald Sie einen Prozedurnamen finden, prüfen Sie
 - ob es sich um einen Befehl oder eine Operation handelt,
 - wie viele Parameter gegebenenfalls erforderlich sind.
2. Das erste Wort in einer Programmzeile ist immer ein Befehl.
3. Eine Operation stellt immer den Parameter einer anderen Prozedur dar.
4. Ein Parameter kann explizit angegeben sein (als Wort beginnend mit einem Anführungszeichen, oder als eine in eckigen Klammern stehende Liste), er kann das Ergebnis einer Operation sein, oder er ist ein mit einem Doppelpunkt beginnender Name.
5. Wenn alle erforderlichen Parameter eines Befehls vorhanden sind, muß die folgende Prozedur ein Befehl sein. Beachten Sie, daß alle zu einer Prozedur gehörenden Parameter in der gleichen Programmzeile stehen müssen.

Hier ein Beispiel für eine komplexe Programmzeile:

```
?WH :N [SEI "I SUMME :I 1 DZ WERT WORT :TOTO :I] DRUCKE  
"HALLO
```

Wir wollen diese Zeile analysieren:

WH ist ein Befehl mit zwei Parametern; einer Zahl und einer Liste mit Anweisungen. DRUCKE ist ein Befehl mit einem Parameter, dem Wort HALLO.

Es gibt also zwei Hauptgruppen:

Gruppe 1: WH und seine Parameter

Gruppe 2: DRUCKE und sein Parameter

Die beiden Parameter von SEI sind :N sowie eine Liste mit Anweisungen.

Gruppe 1.a: :N

Gruppe 1.b: Liste mit Anweisungen

Alle Anweisungen der Liste liegen in einer Programmzeile. SEI ist ein Befehl mit zwei Parametern. DZ ist ein Befehl mit einem Parameter. Weitere Befehle sind in der Liste nicht vorhanden. Damit folgt:

Gruppe 1.b.1: SEI und seine Parameter

Gruppe 1.b.2: DZ und sein Parameter

Der erste Parameter von SEI ist :I, also das Wort I. Der zweite ist SUMME :I 1, also das Ergebnis der Operation SUMME, die selbst zwei Parameter besitzt, nämlich den Inhalt von I sowie die Zahl 1. Wir erhalten:

Gruppe 1.b.1.a: "I

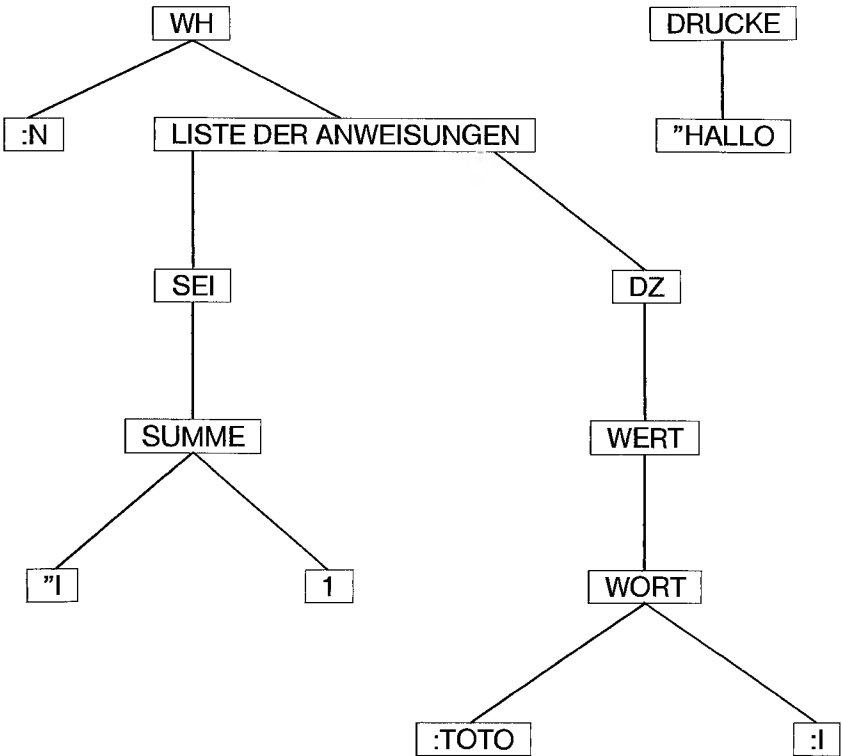
Gruppe 1.b.1.b: SUMME und seine Parameter

Betrachten wir nun die DZ-Gruppe. Der Parameter von DZ ist WERT WORT :TOTO :I, also das Ergebnis der Operation WERT.

WERT ist ein Grundwort, eine Operation, die mit einem einzigen Parameter arbeitet. Dieser Parameter ist in unserem Fall WORT :TOTO :I, also das Ergebnis der Operation WORT.

Die Operation WORT arbeitet mit den beiden Parametern :TOTO und :I, d.h., den Objekten TOTO und I.

Damit ist unsere Analyse beendet. Wir können die Ergebnisse in einem Diagramm zusammenfassen:



Kapitel I/6

Sonderzeichen

In einer Logo-Programmzeile sind bestimmte Zeichen für spezielle Aufgaben reserviert.

Die Separatoren:

Leerzeichen dienen zur Unterscheidung einzelner Wörter.

Eckige Klammern [] kennzeichnen den Beginn (eckige Klammer auf) und das Ende (eckige Klammer zu) einer Liste.

Runde Klammern () kennzeichnen den Beginn oder das Ende einer Zeichen- oder Anweisungsgruppe, die insgesamt mit besonderer Priorität behandelt werden soll (s. Kapitel II/5).

Das Dollarzeichen \$ bewirkt die Nichtbeachtung des Separators Leerzeichen, vor dem das Dollarzeichen steht.
Beispiel:

```
?DZ "GUTEN$ TAG  
GUTEN TAG
```

Die Operatoren:

/ Zeichen für die Division

* Zeichen für die Multiplikation

- Zeichen für die Subtraktion, Vorzeichen für negative Zahlen, Operator zur Umkehr eines Wertes.

Beispiel: wenn :X den Wert 2 hat, liefert -:X den Wert -2.

+ Zeichen für die Addition

> „größer als“-Zeichen

< „kleiner als“-Zeichen

= Gleichheitszeichen

Sonstige Sonderzeichen:

Anführungszeichen (") werden unmittelbar vor ein Wort gesetzt, wenn dieses als Parameter einer Prozedur dient. Zahlen benötigen keine Anführungszeichen.

Doppelpunkte (:) werden unmittelbar vor ein Wort gesetzt, wenn dieses einen Namen darstellt, dem ein bestimmter Inhalt zugewiesen werden soll. Beispiel: DZ :X entspricht DZ WERT "X"

TEIL II

Die Grundwörter der Programmiersprache Logo

In den folgenden Kapiteln werden alle Grundwörter vorgestellt und ausführlich besprochen. Wir haben sie nach Anwendungsgebieten zusammengefaßt.

Jedes Grundwort wird in folgender Weise vorgestellt:

Name des Grundworts, Parameter Anweisungstyp
Abkürzung/Bedeutung des Namens

Beispiel:

DZ Term		Befehl
(Drucke Zeile)		
	Funktionsbeschreibung	
	Beispiel	

Bei Operationen geben wir unter der Anweisungsart zusätzlich die Art des Ergebnisses an. Dabei benutzen wir folgende Abkürzungen:

n = Zahl

Z = Zeichen

Adr = Speicheradresse

Log = Logisches Ergebnis eines Vergleichs (WAHR oder FALSCH)

Bestimmte Grundwörter erwarten ganzzahlige Werte als Parameter. Wenn der übergebene Wert nicht ganzzahlig ist, wird automatisch nur der ganzzahlige Teil benutzt.

Nur die arithmetischen Grundwörter akzeptieren Zahlen größer als 65535 und kleiner als -65535.

Die mit einem Punkt beginnenden Grundwörter sind mit besonderer Vorsicht zu benutzen.

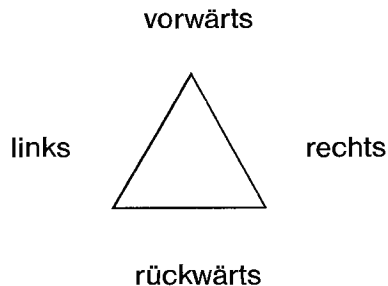
Die in diesem Handbuch definierten Prozeduren sind nicht nur Beispiele für die Anwendung der Grundwörter, sondern gleichzeitig Hilfsmittel, die Ihnen die Programmierung erleichtern können. Ein Ausrufungszeichen (!) am Ende einer Programmzeile zeigt an, daß die Zeile mit der Folgezeile fortgesetzt wird.

Kapitel II/1

Die Welt des Igels

Wir behandeln in diesem Kapitel die Grundwörter zur Steuerung des Igels, des Schreibstifts sowie zur Anpassung des Igefeldes.

Der Igel ist das auf dem Bildschirm sichtbare kleine Dreieck. Sie können ihn sich als eine Art Zeichenroboter vorstellen, der mit einem Schreibstift ausgerüstet ist und seinen Weg auf dem Bildschirm durch eine Linie oder Spur markieren kann. Die Spitze des Dreiecks kennzeichnet die „Blickrichtung“ des Igels.



Der momentane „Zustand“ des Igels wird bestimmt durch seine **Position** im Feld und seine **Richtung**.

Sobald Sie eines der Grundwörter VW, RW, RE, LI, SKURS, SO, VI, ZI, MITTE, TB, SHGF, FEN, RS, RAND oder PUNKT aufrufen, wird der Grafikbereich (Igefild) sichtbar. Um unteren Bildschirmrand verbleibt ein vier Zeilen umfassender Textbereich.

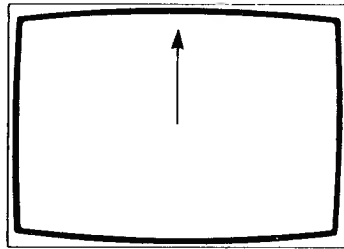
Der Grafikbereich verschwindet, sobald der Logo-Editor aufgerufen wird.

1.1 Die Bewegungen

VORWAERTS n
VW n

(Befehl)

bewegt den Igel um n Schritte (maximal 32767) vorwärts. Nur der ganzzahlige Teil von n wird benutzt. Die für n erlaubten Werte sind abhängig vom Zustand des Igels und des Feldes (s. Abschnitt 1.2).

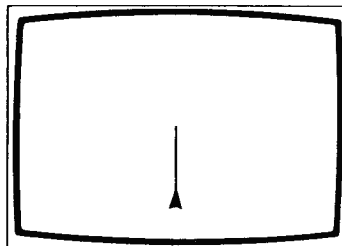


VORWAERTS 50

RUECKWAERTS n
RW n

(Befehl)

bewegt den Igel um n Schritte (maximal 32767) rückwärts. Nur der ganzzahlige Teil von n wird benutzt. Die für n erlaubten Werte sind abhängig vom Zustand des Igels und des Feldes (s. Abschnitt 1.2).

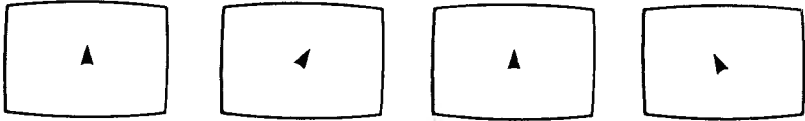


RUECKWAERTS 50

RE n
(Rechts)

(Befehl)

dreht den Igel um n Grad nach rechts (im Uhrzeigersinn). n ist eine Zahl zwischen -360 und 360. Nur der ganzzahlige Teil von n wird benutzt.



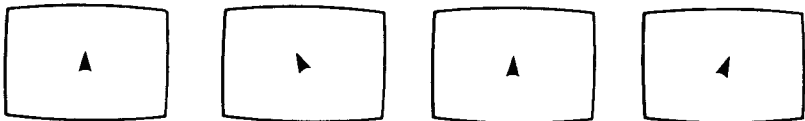
RE 45

RE -45

LI n
(Links)

(Befehl)

dreht den Igel um n Grad nach links (gegen den Uhrzeigersinn). n ist eine Zahl zwischen -360 und 360. Nur der ganzzahlige Teil von n wird benutzt.



LI 45

LI -45

Folgende Prozeduren zeichnen ein Kreissegment (rechts und links herum):

```
PR RSEGMENT :N  
WH :N [VW 1 RE 1]  
ENDE
```

```
PR LSEGMENT :N  
WH :N [VW 1 LI 1]  
ENDE
```

Einen Kreis mit beliebigem Umfang ermöglicht folgende Prozedur:

PR KREIS :SCHRITTW
WH 360 [VW :SCHRITTW RE 1]
ENDE

ORT

(Operation)
(Liste)

liefert eine aus zwei Zahlen bestehende Liste. Die erste Zahl ist die horizontale, die zweite Zahl die vertikale Koordinate des Igels in seinem Feld (s. Abschnitt 1.2).

Die horizontale Koordinate erhalten Sie mit:

PR XKO
RG ERSTES ORT
ENDE

Die vertikale Koordinate erhalten Sie mit:

PR YKO
RG LETZTES ORT
ENDE

Das Koordinatensystem für den Grafikbereich hat folgenden Aufbau:

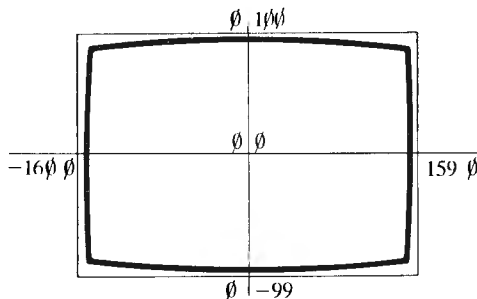


Bild S. 32

SO Liste
(Setze Ort)

(Befehl)

bewegt den Igel an die in der Liste angegebene Position. Die **Position** wird angegeben durch einen horizontalen und einen vertikalen Koordinatenwert. SO ändert nicht die Blickrichtung des Igels. Das Koordinatenpaar ist vom Zustand des Grafikbereichs (Igel-feld) abhängig (s. Abschnitt 1.2).

Es wird nur der ganzzahlige Teil der Koordinatenwerte benutzt.

Ein Viereck erhalten Sie beispielsweise mit:

```
PR VIERECK
SH
SO [-60 -60]
SA
SO [-60 60 60 60 -60 -60 -60]
ENDE
```

Bewegung parallel zur horizontalen Koordinatenachse:

```
PR HOR :X
SO LISTE :X LETZTES ORT
ENDE
```

Bewegung parallel zur vertikalen Koordinatenachse:

```
PR VERT :Y
SO LISTE ERSTES ORT :Y
ENDE
```

KURS

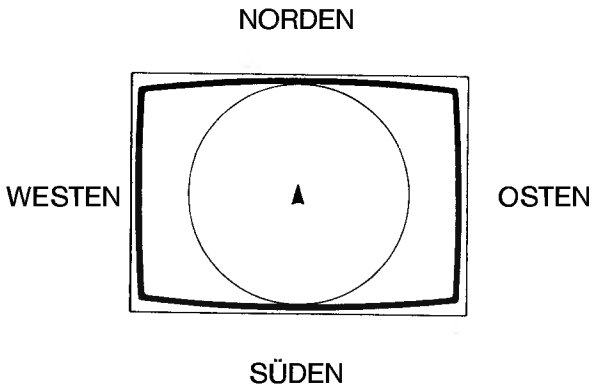
(Operation)
(n)

liefert die Blickrichtung des Igels in Grad (eine Zahl zwischen 0 und 359).

SKURS n
(Setze Kurs)

(Befehl)

gibt dem Igel die durch „n Grad“ definierte Blickrichtung (Norden entspricht dem oberen Bildschirmrand). Beachten Sie, daß RE und LI zwar auch die Blickrichtung des Igels ändern, aber immer relativ zu seinem Ausgangszustand. SKURS legt dagegen eine absolute Richtung fest, wobei n Werte zwischen -360 und 360 (Grad) annehmen darf. Für n wird nur der ganzzahlige Teil des Wertes benutzt.



1.2 Das Igelfeld

Die Koordinaten der Igelposition beziehen sich auf ein kartesisches Koordinatensystem, dessen Mittelpunkt (0,0) in der Mitte des Bildschirms liegt.

Nach dem Einschalten bildet der Bildschirm das Feld des Igels, d.h., dem Igel kann keine Position zugewiesen werden, die außerhalb des Bildschirms liegt. Zum Feldbereich gehört nicht nur der Grafikbereich, sondern auch der standardmäßig vier Zeilen umfassende Textbereich im unteren Teil des Bildschirms. Das Igelfeld kann vergrößert werden (s. FEN).

Der sichtbare Feldbereich ist von einem Rand umgeben, dessen Farbe geändert werden kann (s. SR).

Einige technische Anmerkungen:

Der Bildschirm besteht aus 200 Grafikzeilen, von denen jede in 40 Segmente zu je 8 Punkten unterteilt ist. In jedem Segment können die Punkte entweder die Farbe des Schreibstiftes oder die Farbe des Hintergrundes annehmen.

Der Bildschirm bildet den Inhalt eines bestimmten Speicherbereiches des Computers ab, und zwar ist immer einem Byte des Speichers ein Segment des Bildschirms zugeordnet.

Die ersten acht Bildpunkte in der linken oberen Ecke des Bildschirms werden gesteuert durch die acht Bits des Bytes mit der Adresse 16384 (TO7-70) bzw. 0 (MO5 E). Nacheinander ist so jeweils einem Segment ein Byte zugeordnet. Das Byte für die rechte untere Ecke des Bildschirms hat die Adresse 24383 (TO7-70) bzw. 7999 (MO5 E).

Die Farbe der Bildpunkte wird durch die Bits des entsprechenden Bytes im Bildschirmspeicher festgelegt. Ein Bit kann den Wert 0 oder 1 annehmen. Der Wert 0 entspricht der Hintergrundfarbe, der Wert 1 der jeweiligen Farbe des Schreibstiftes. Durch ein Byte können nicht mehr als zwei Farben festgelegt werden, denn die jeweilige Zeichensstofffarbe gilt für alle Bits mit dem Wert 1. Um dem System mitzuteilen, ob der Farb- oder der Zeichenspeicher benutzt werden soll, wird das niederwertige Bit des Bytes mit der Adresse 59331 (TO7-70) bzw. 42944 (MO5 E) entweder auf 0 oder auf 1 gesetzt. Die höherwertigen Bits dürfen nicht verändert werden, da sie zur Steuerung der Randfarbe, der Tastatur und der Grafikseiten dienen.

HF

(Operation)

(Hintergrundfarbe)

liefert eine ganze Zahl, die den Farbcode für die Feldfarbe darstellt. Beispiel:

?DZ HF

2

SHGF n
(Setze Hintergrundfarbe)

(Befehl)

definiert die Farbe n des Igefildes, wobei n der Farbcode ist.

n Farbe

0 Schwarz	8 Grau
1 Rot	9 Blassrot
2 Grün	10 Blassgrün
3 Gelb	11 Blassgelb
4 Blau	12 Blassblau
5 Violett	13 Blasslila
6 Hellblau	14 Blasshellblau
7 Weiß	15 Orange

?SHGF 2 – das Feld wird grün.

SR n
(Setze Randfarbe)

(Befehl)

definiert die Randfarbe, wobei n der Farbcode ist (s.o.).
Die folgende Prozedur ermöglicht es, die Randfarbe durch einen
Tastendruck zu ändern:

```
PR RANDFARBE :X  
SR :X  
DZ LZ  
RANDFARBE :X + 1  
ENDE
```

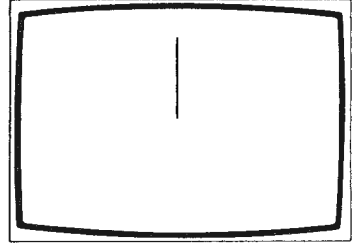
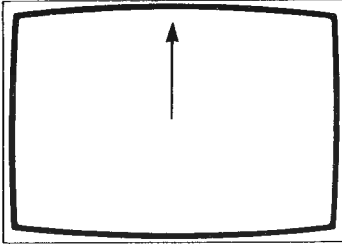
Um die Randfarbe der Bildschirmfarbe anzupassen, können Sie
oben die 4. Zeile löschen und dann folgende Prozedur benutzen:

```
PR WAHL :N  
SHGF :N  
RANDFARBE N  
WAHL :N + 1  
ENDE
```

VI
(Verstecke Igel)

(Befehl)

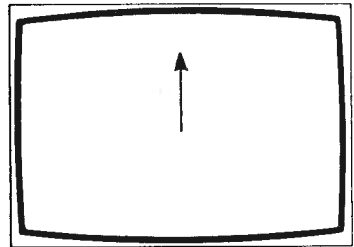
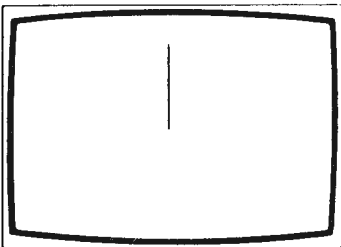
macht den Igel unsichtbar, ohne dabei seinen Zustand zu ändern.
Die Ausführung von Zeichnungen wird dadurch beschleunigt.



ZI
(Zeige Igel)

(Befehl)

macht den Igel sichtbar, ohne dabei seinen Zustand zu ändern (er kann allerdings nur gesehen werden, wenn er sich innerhalb der Bildschirmgrenzen befindet). Nach dem Einschalten oder nach Ausführung des Befehls LS ist der Igel automatisch sichtbar.



SICHTBAR?

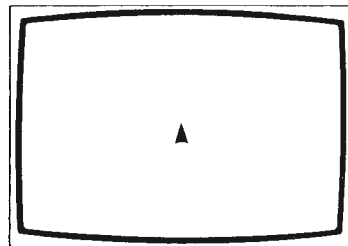
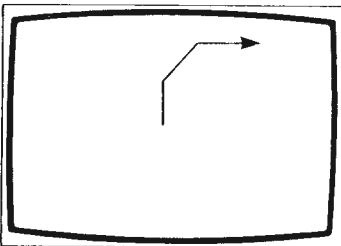
(Operation)
(Log)

liefert WAHR, wenn der Igel sichtbar ist, und sonst FALSCH.

LS (Lösche Schirm)

(Befehl)

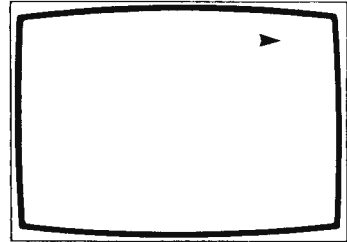
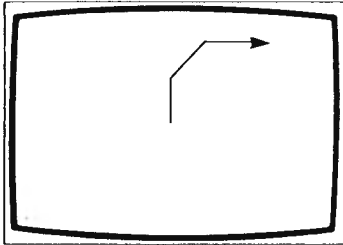
versetzt den Bildschirm in seinen Ausgangszustand. Der Grafikbereich wird gelöscht und in blauer Hintergrundfarbe dargestellt, der Textbereich umfaßt vier Zeilen, und der Igel sitzt mit Blickrichtung Norden und gesenktem Schreibstift (s. Abschnitt 1.3) im Koordinatenmittelpunkt (0,0).



LB
(Lösche Bild)

(Befehl)

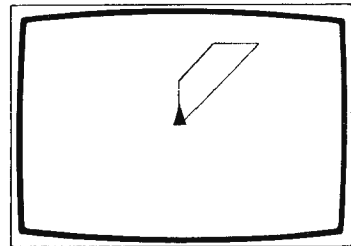
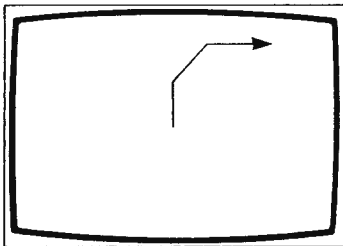
löscht das Feld, ohne den Zustand des Igels oder des Schreibstifts (s. Abschnitt 1.3) zu beeinflussen.



MITTE

(Befehl)

verschiebt den Igel in die Bildschirmmitte und dreht ihn in Richtung Norden, ohne dabei den Zustand des Schreibstifts (s. Abschnitt 1.3) zu beeinflussen. MITTE entspricht der Befehlsfolge `SO [0 0] SKURS 0.`



MAB
(Maßstab)

(Operation)
(Liste)

liefert eine Liste mit zwei ganzen Zahlen. Die erste repräsentiert die hundertfach vergrößerte horizontale Komponente der Schrittweite des Igels, die zweite die hundertfach vergrößerte vertikale Komponente. Die Komponenten ergeben sich durch die Projektion der Schrittweite auf die beiden Koordinatenachsen. Nach dem Einschalten haben beide Listenelemente den Wert 100.

SMAB *Liste*
(Setze Maßstab)

(Befehl)

definiert die horizontale und vertikale Komponente der Schrittweite des Igels bei Ausführung der Befehle VW oder RW (s. SCALA). Beide Komponenten müssen in der **Liste** angegeben sein. Ihr Wert darf zwischen 0 und 200 liegen. Die festgelegten Komponenten werden durch eine Änderung des Feldzustands (LS, FEN, usw.) nicht beeinflusst.

```
?SMAB [150 180]  
?DZ SMAB  
150 180
```

Die Änderung der Komponenten der Schrittweite bietet sowohl die Möglichkeit, bestimmte geometrische Muster zu verzerren (Darstellung eines Kreises als Ellipse, eines Quadrats als Viereck, usw.), als auch die Möglichkeit, Zeichnungen zu vergrößern oder zu verkleinern.

Benutzen wir als Beispiel die schon bei der Beschreibung des LI-Befehls definierte Prozedur KREIS. Mit Hilfe des SSCALA-Befehls können wir mit dieser einen Prozedur eine Anzahl ineinanderliegender Kreise erzeugen, die sich alle an derselben Stelle berühren:

```
PR FAECHER  
SMAB [50 50] KREIS .5  
SMAB [100 100] KREIS .5  
SMAB [150 150] KREIS .5  
SMAB [200 200] KREIS .5  
ENDE
```

Umformung eines Kreises in zwei Ellipsen:

```
PR ELLIPSEN  
SMAB [20 100] KREIS 1  
SMAB [100 20] KREIS 1  
ENDE
```

RAND

(Befehl)

begrenzt das Igelfeld auf den Bildschirmbereich. Die vertikale Koordinate des Igels muß anschließend zwischen -99 und 100, die horizontale Koordinate zwischen -160 und 159 liegen. Durch den RAND-Befehl wird der Grafikbereich gelöscht und neu initialisiert. Nach dem Einschalten ist RAND automatisch aktiv.

RS

(Befehl)

(Rückkehr auf gleicher Spur)

bewirkt die Rückkehr des Igels in den Grafikbereich. Die vertikale Koordinate des Igels liegt immer zwischen -99 und 100, die horizontale Koordinate immer zwischen -160 und 159. Sobald der Igel den Grafikbereich nach oben verläßt, tritt er unten wieder ein. Entsprechendes gilt für den linken und rechten Rand. RS löscht und initialisiert den Grafikbereich.

FEN

(Befehl)

(Fenster)

erweitert das Igelfeld. Die horizontale und vertikale Koordinate des Igels darf anschließend zwischen -32768 und 32767 liegen. Der Bildschirm stellt jetzt nur noch ein Fenster dar, durch das ein Ausschnitt des Feldes zu sehen ist. Der Igel kann sich also außerhalb des sichtbaren Bereichs befinden. Sobald der Igel das vergrößerte Feld verläßt, tritt er auf der entgegengesetzten Seite wieder ein. FEN löscht und initialisiert den Grafikbereich.

Die folgenden Prozeduren dienen dazu, den Igel vom Bildschirmrand abprallen zu lassen, nachdem der Befehl FEN gegeben wurde:

PR ABPRALL :N
WENN GLEICH? :N 0 [RK]
VW 1
WENN AUSSEN ORT [SKURS 360 - KURS]
ABPRALL :N - 1
ENDE

PR AUSSEN :L
WENN KLEINER? 159 ERSTES :L [RG WAHR]
WENN GROESSER? -160 ERSTES :L [RG WAHR]
WENN GROESSER? -99 LETZTES :L [RE 180 R, G WAHR]
WENN KLEINER? 100 LETZTES :L [RE 180 RG, WAHR]
RG FALSCH
ENDE

PUNKT *Liste*

(Befehl)

bewirkt die Ausgabe der Punkte, deren Koordinaten in der Liste angegeben sind. Alle Punkte werden in der aktuellen Farbe des Zeichenstiftes dargestellt. Der Zustand des Iglers bleibt dabei unverändert. Die horizontalen Punktkoordinaten dürfen zwischen -160 und 159, die vertikalen zwischen -99 und 100 liegen. Es werden nur die ganzzahligen Teile der Koordinatenwerte benutzt.

PR SPIRALPUNKTE :X
WENN :X > 80 [SH MITTE RK]
SH VW :X PUNKT 0 RT RE 70
SPIRALPUNKTE :X + 2
ENDE

1.3 Der Zeichenstift

Der **Zustand** des Zeichen- oder Schreibstifts wird bestimmt durch die jeweilige **Situation** (Stift gehoben oder gesenkt) und die aktuelle **Farbe**.

SA (Befehl)
(Stift ab)

senkt den Stift, damit der Igel auf dem Bildschirm zeichnen kann, d.h., er hinterläßt bei einer Bewegung eine Spur in der aktuellen Farbe. Nach dem Einschalten oder einer Initialisierung des Grafikbereichs ist der Stift gesenkt und die Zeichenfarbe ist Hellblau.

SH (Befehl)
(Stift hoch)

hebt den Stift, so daß der Igel bei einer Bewegung keine Spur hinterläßt. Die Zeichenfarbe bleibt unverändert.

SA? (Operation)
(Stift ab?) (Log)

– liefert WAHR, wenn der Zeichenstift gesenkt ist, und sonst FALSCH.

SF (Operation)
(Stiftfarbe) (n)

liefert eine Zahl, die den aktuellen Farbcode der Zeichenfarbe darstellt.

?SSF 6
?DZ SF
6

SSF n (Befehl)
(Setze Stiftfarbe)

definiert die Stiftfarbe **n**, wenn **n** eine positive Zahl oder Null ist. **n** ist der Farbcode. Für ganzzahlige **n** zwischen 0 und 15 gelten folgende Zuordnungen:

Stiftfarbe	Farbcode		
Schwarz	0	Grau	8
Rot	1	Blassrot	9
Grün	2	Blassgrün	10
Gelb	3	Blassgelb	11
Blau	4	Blassblau	12
Violett	5	Blassviolett	13
Hellblau	6	Blasshellblau	14
Weiß	7	Orange	15

Wenn n negativ ist, wird die Hintergrundfarbe des Segments benutzt, durch das sich der Igel gerade bewegt.

Wenn n ganzzahlig ist und zwischen -1 und -16 liegt, gilt folgende Codetabelle:

Stiftfarbe	Farbcode		
Schwarz	-1	Grau	-9
Rot	-2	Blassrot	-10
Grün	-3	Blassgrün	-11
Gelb	-4	Blassgelb	-12
Blau	-5	Blassblau	-13
Violett	-6	Blassviolett	-14
Hellblau	-7	Blasshellblau	-15
Weiß	-8	Orange	-16

Wenn Sie z.B. 50 Schritte vorwärts gegangen sind und die dabei erzeugte Spur anschließend löschen wollen, müssen Sie den Weg mit der Hintergrundfarbe als Stiftfarbe zurückgehen. Dabei können Ihnen folgende Prozeduren behilflich sein:

```
PR RADIEREN
SEI "FARBE SF
SSF - (HF + 1)
ENDE
```

```
PR RADIERHALT
SSF :FARBE
ENDE
```

?VW 50
?RADIEREN
?RW 50
?RADIERT
?RW 30

Die folgende Prozedur zeigt Ihnen ein Beispiel aller zur Verfügung stehenden Stiftfarben:

PR MUSTER :X
SSF :X
WH 10 [VW 50 RE 90 VW 0.5 LI 90 RW 50 RE
90 VW 0.5 LI 90]
MUSTER :X + 1
ENDE

Kapitel II/2

Die Welt der Wörter und der Listen

Es gibt zwei Arten von Logo-Termen: **Wörter** und **Listen**. Die Grundwörter, wir Ihnen in diesem Kapitel vorstellen, dienen dazu, Wörter und Listen zu prüfen, aufzubauen oder zu verkleinern.

Die Wörter

Ein Wort besteht aus Zeichen. Wörter in einer Logo-Programmzeile sind durch Separatoren (meistens Leerzeichen) voneinander getrennt. Ein Wort kann in bestimmten Fällen mit einem Doppelpunkt (:) oder einem Anführungszeichen (") beginnen.

Das Dollarzeichen (\$) dient dazu, Wörter zu definieren, die Leerzeichen enthalten (siehe Kapitel I/6).

Wörter können auch aus nur einem Zeichen bestehen. Es gibt sogar ein spezielles Wort, das kein Zeichen enthält. Es heißt **Leerwort** und wird in einer Logo-Programmzeile dargestellt durch ein Anführungszeichen, gefolgt von einem Separator (ein Doppelpunkt, gefolgt von einem Separator, kennzeichnet das Objekt eines Leerwortes).

Die Listen

Eine Liste besteht aus einer Anzahl von Logo-Termen (Wörtern oder anderen Listen). Die Terme sind die Elemente der Liste.

Eine Liste ist in einer Logo-Programmzeile in eckigen Klammern eingeschlossen. Enthält die Liste keine Elemente, so handelt es sich um die leere Liste [].

2.1 Prüfen

GLEICH? *Term1 Term2*
Term1 = Term2

(Operation)
(Log)

liefert WAHR, wenn *Term1* und *Term2* identische Wörter oder Listen sind, und sonst FALSCH.

In der Normalform ($Term1 = Term2$) müssen die beiden Parameter ein Gleichheitszeichen (=) einschließen. Die Definition der Normalform finden Sie in Kapitel II/7.

?DZ GLEICH? "HALLO "HALLO
WAHR

?DZ GLEICH? [HALLO] [[HAL]LO]
FALSCH

Hinweis: Zwei Listen sind identisch, wenn alle Elemente identisch sind. Zwei Wörter sind identisch, wenn sie aus der gleichen Zeichenfolge bestehen (ausgenommen Zahlen). Zwei Zahlen sind identisch, wenn sie den gleichen Wert darstellen.

?DZ WORT 1 "E2
1E2

?DZ 100
100

?DZ GLEICH? WORT 1 "E2 100
WAHR

LEER? *Term*

(Operation)
(Log)

liefert WAHR, wenn der *Term* die leere Liste oder das Leerwort ist, und sonst FALSCH.

?DZ LEER? []
WAHR

LISTE? Term(Operation)
(Log)liefert WAHR, wenn der *Term* eine Liste ist, und sonst FALSCH.?DZ LISTE? "HALLO
FALSCH?DZ LISTE? [1 2 3 4 5]
WAHR**WORT? Term**(Operation)
(Log)liefert WAHR, wenn der *Term* ein Wort ist, und sonst FALSCH.
Die Operation WORT? *Term* entspricht der Operation NICHT LISTE? *Term*.?DZ WORT? "JAWOHL
WAHR**EL? Term Liste
(Element?)**(Operation)
(Log)

liefert WAHR, wenn der Term ein Element der Liste ist, und sonst FALSCH.

?DZ EL? „GUT [DAS WETTER IST GUT]
WAHR

Die folgende Prozedur prüft, ob ein bestimmter Buchstabe in einem Wort enthalten ist:

PR WORTELEMENT :BUCHST :WORT
WENN LEER? :WORT [RG "FALSCH]
WENN GLEICH? :BUCHST ERSTES :WORT [RG "
WAHR]RG WORTELEMENT :BUCHST OE :WORT
ENDE?DZ WORTELEMENT "Z "SCHWARZ
WAHR

Um herauszufinden, ob ein Wort nur in einer von zwei Listen enthalten ist, ist folgende Prozedur geeignet:

```
PR VERGLEICH :LISTE1 :LISTE2 :WORT
WENN UND? EL? :WORT :LISTE1 NICHT? EL?
:WORT :LISTE2 [DZ WAHR] [DZ FALSCH]
ENDE
```

```
?VERGLEICH [ICH DU ER SIE ES] [ICH DU E]
R] "DU
FALSCH
```

ZAHL? Term

(Operation)
(Log)

liefert WAHR, wenn der *Term* eine Zahl ist, und sonst FALSCH.

```
?DZ ZAHL? [DREI]
FALSCH
```

```
?DZ ZAHL? 3
WAHR
```

ASCII Wort

(Operation)
(n)

liefert den ASCII-Code des ersten Zeichens im *Wort*. Wenn das *Wort* das Leerwort ist, wird der Code 0 übergeben. Sie finden die ASCII-Codes in Anhang B.

```
?DZ ASCII "A
65
```

Mit folgender Prozedur wird ein Wort codiert:

```
PR WORTCODE :WORT
WENN LEER? :WORT [RG[]] [RG ME ASCII ER
STES :WORT WORTCODE OE :WORT]
ENDE
```

```
?DZ WORTCODE "HANS
72 65 78 83
```

Damit läßt sich nun auch ein ganzer Satz verschlüsseln:

```
PR SATZCODE :MELDUNG
WENN LEER? :MELDUNG [RG " ] [RG SATZ
WORTCODE ERSTES :MELDUNG SATZCODE OE
:MELDUNG]
ENDE
```

```
?DZ SATZCODE [HANS IST IM HAUS]
72 65 78 83 73 83 84 73 77 72 65 85 83
```

ZEICHEN n (Operation)
(Z)

liefert das Zeichen mit dem ASCII-Code n . n muß eine positive ganze Zahl sein. Ist n größer als 255, wird sie modulo 256 benutzt. Für $n = 0$ gibt ZEICHEN das Leerwort zurück.

Liegt n im Bereich $1 < n < 31$, so liefert ZEICHEN ein nicht darstellbares Steuerzeichen. Für $128 < n < 255$ wird ein frei definierbares Zeichen zurückgegeben (s. Anhang B).

```
?DZ ZEICHEN 36
$
```

```
PR ALPHABET :VON :BIS :LISTE
WENN GROESSER? :VON :BIS [RG :LISTE]
RG ALPHABET SUMME 1 :VON :BIS MT ZEICHE
N :VON :LISTE
ENDE
```

```
DZ ALPHABET 65 90 []
A B C D E F ..... Z
```

LAENGE *Liste* (Operation)
(n)

liefert die Anzahl der Elemente in der Liste.

```
?DZ LAENGE [ANZAHL DER WORTE?]
3
```

Folgende Prozedur zählt die Anzahl der Buchstaben in einem Wort:

```
PR ZAEHLEN :WORT
WENN LEER? :WORT [RG 0] [RG 1 + ZAEHLEN!
OE :WORT]
ENDE
```

```
?DZ ZAEHLEN "BUCHSTABENZAEHLPROZEDUR
23
```

2.2 Abbauen

ERSTES *Term*

(Operation)
(Term)

liefert das erste Element, wenn der **Term** eine Liste ist. Handelt es sich bei dem **Term** um ein Wort, so übergibt ERSTES das erste Zeichen des Wortes. ERSTES darf weder auf die leere Liste noch auf das Leerwort angewendet werden.

```
?DZ ERSTES "HALLO
H
```

```
? DZ ERSTES [HALLO UND GUTEN TAG]
HALLO
```

OE *Term*
(Ohne Erstes)

(Operation)
(Term)

liefert die Liste ohne das erste Element, wenn der Term eine Liste ist. Handelt es sich bei dem Term um ein Wort, so übergibt OE das Wort ohne das erste Zeichen. OE darf weder auf die leere Liste noch auf das Leerwort angewendet werden.

```
?DZ OE "HALLO
ALLO
```

```
?DZ OE [HALLO UND GUTEN TAG]
UND GUTEN TAG
```

LETZTES Term(Operation)
(Term)

liefert das letzte Element, wenn der Term eine Liste ist. Handelt es sich bei dem Term um ein Wort, so übergibt LETZTES das letzte Zeichen des Wortes. LETZTES darf weder auf die leere Liste noch auf das Leerwort angewendet werden.

```
?DZ LETZTES "HALLO
0
```

OL Term
(Ohne Letztes)(Operation)
(Term)

liefert die Liste ohne das letzte Element, wenn der Term eine Liste ist. Handelt es sich bei dem Term um ein Wort, so übergibt OL das Wort ohne das letzte Zeichen. OL darf weder auf die leere Liste noch auf das Leerwort angewendet werden.

```
?DZ OL "HALLO
HALL
```

```
?DZ OL [HALLO UND GUTEN TAG]
HALLO UND GUTEN
```

Mit Hilfe der Operation OL läßt sich ein Wort umkehren:

```
PR UMKEHR :WORT
WENN LEER? WORT [DZ][RK]
DRUCKE LETZTES :WORT UMKEHR OL :WORT
ENDE
```

```
?UMKEHR "HALLO
OLLAH
```

EL n Liste
(Element)(Operation)
(Term)

liefert das durch n angegebene Element aus der *Liste*. n muß ganzzahlig, positiv, kleiner oder gleich der Anzahl der Elemente sein.

?DZ EL 2 [ANZAHL DER WORTE?]
DER

Mit EL läßt sich beispielsweise ein Zufallswort aus einer Liste wählen:

PR ZWAHL :LISTE
DZ EL ZZ 1 + LAENGE :LISTE :LISTE
ENDE

?ZWAHL [HANS MARIA PETER CLAUDIA]
PETER

Die folgende Prozedur bestimmt die Position in einem Wort, an der ein Zeichen zum ersten Mal auftritt:

PR POS1 :ZEICHEN :WORT
WENN LEER? :WORT [RG [NICHT ENTHALTEN]]
WENN GLEICH? ERSTES :WORT :ZEICHEN [RG !
1] [RG SUMME 1 POS1 :ZEICHEN OE :WORT]
ENDE

?DZ POS1 "A "BAHNFAHRT
2

2.3 Aufbauen

WORT *Wort1 Wort2*

(Operation)
(Wort)

liefert ein Wort, das aus *Wort1* und *Wort2* besteht, d.h., *Wort1* und *Wort2* werden verknüpft. Bei der Verknüpfung eines Wortes mit dem Leerwort bleibt das Wort unverändert.

?DZ WORT "BAHN "HOF
BAHNHOF

SATZ *Term1 Term2*(Operation)
(Liste)

liefert eine aus den beiden Termen bestehende Liste. Beispielsweise erzeugt SATZ [GUTEN] [TAG] die Liste [GUTEN TAG].

?DZ SATZ "HALLO [UND GUTEN TAG]
HALLO UND GUTEN TAG

Wenn beide *Terme* Listen sind, verknüpft SATZ beide Listen so, daß in der resultierenden Liste zuerst die Elemente des ersten und dann die Elemente des zweiten *Terms* stehen.

Wenn beide *Terme* Wörter sind, wirkt SATZ wie LISTE (s. LISTE).

Wenn *Term1* ein Wort und *Term2* eine Liste ist, wirkt SATZ wie ME (s. ME).

Wenn *Term1* eine Liste und *Term2* ein Wort ist, wirkt SATZ *Term1 Term2* wie MT *Term2 Term1*.

SATZ [DIE SONNE] [SCHEINT] liefert [DIE SONNE SCHEINT]

SATZ "ROTER "MOND liefert [ROTER MOND]

SATZ "DIE [SONNE SCHEINT] liefert [DIE SONNE SCHEINT]

SATZ [MEIN HUND] "BELLO liefert [MEIN HUND BELLO]

LISTE *Term1 Term2*(Operation)
(Liste)

liefert eine Liste, deren erstes Element *Term1* und deren zweites Element *Term2* ist.

LISTE [HEUTE SCHEINT] [DIE SONNE] liefert [[HEUTE SCHEINT]
[DIE SONNE]]

Damit lassen sich Listen erweitern:

PR EINSETZEN :WORT
SEI "LISTE MT :WORT :LISTE
ENDE

?SEI "LISTE [GRUNDWORTE DEUTSCH-ENGLISCH:]
?EINSETZEN [VW FORWARD]
?DZ :LISTE
GRUNDWORTE DEUTSCH-ENGLISCH:[VW FORWARD]

ME *Term Liste*
(Mit Erstem)

(Operation)
(Liste)

liefert die angegebene Liste mit dem angegebenen *Term* als erstem Element.

ME "A [] liefert die Liste [A]

ME [PARIS IST] [EINE STADT] liefert [[PARIS IST] EINE STADT]

MT *Term Liste*
(Mit Letztem)

(Operation)
(Liste)

liefert die angegebene Liste mit dem angegebenen *Term* als letztem Element.

MT [BELLO] [MEIN HUND] liefert die Liste [MEIN HUND[BELLO]]

Kapitel II/3

Die logischen Grundoperationen

Wir behandeln in diesem Kapitel die bekannten logischen Operationen, die mit den logischen Werten (auch Wahrheitswerte genannt) WAHR und FALSCH arbeiten. Diese Werte sind die Parameter der entsprechenden Grundwörter. Die mit logischen Werten arbeitenden Grundwörter (logische Grundoperationen) liefern das Wort WAHR, wenn der logische Vergleich wahr ist, und das Wort FALSCH, wenn er falsch ist.

Außerhalb dieses Kapitels beschriebene logische Grundwörter finden Sie mit Hilfe der folgenden Tabelle:

<u>Grundwort</u>	<u>Kapitel</u>
EL?	II/2
GLEICH? (=)	II/2
GROESSER? (>)	II/7
GW?	II/4
KLEINER? (<)	II/7
KNOPF?	II/9
KONTACT?	II/9
LISTE?	II/2
LEER?	II/2
NAME?	II/6
PR?	II/4
SA?	II/1
SICHTBAR?	II/1
TASTE?	II/9
WORT?	II/2
ZAHL?	II/2

UND? Log1 Log2

(Operation)
(Log)

liefert WAHR, wenn die logischen Ausdrücke Log1 und Log2 beide wahr sind, und sonst FALSCH.

?DZ UND? GLEICH? 3 (2 + 1) GLEICH? 8 (!
2 * 4)
WAHR

?DZ ALLE? "UND "FALSCH
FALSCH

ODER? Log1 Log2

(Operation)
(Log)

liefert WAHR, wenn entweder der logische Ausdruck Log1 oder der logische Ausdruck Log2 wahr ist (oder wenn beide wahr sind), und sonst FALSCH.

?DZ ODER? GLEICH? 3 (2 + 1) GLEICH? 8 !
(2 * 3)
WAHR

Damit der Igel in seinem Feld bleibt, können Sie folgende Prozedur benutzen:

PR NACH-INNEN
WENN ODER? ERSTES ORT > 80 ERSTES ORT ! < -80 [RE 90]
WENN ODER? LETZTES ORT < -80 LETZTES O!
RT > 80 [RE 90]
VW 1
NACH-INNEN
ENDE

NICHT? *Log*

(Operation)
(Log)

liefert den umgekehrten logischen Wert des Ausdrucks *Log*, also WAHR, wenn *Log* falsch ist, und FALSCH, wenn *Log* wahr ist.

?DZ NICHT? GLEICH? (4 + 1) (21 - 6)
WAHR

WAHR

(Operation)
(Log)

liefert immer das Wort WAHR. Daher muß das Wort WAHR nicht mit einem Anführungszeichen beginnen.

?DZ WAHR
WAHR

FALSCH

(Operation)
(Log)

liefert immer das Wort FALSCH. Daher muß das Wort FALSCH nicht mit einem Anführungszeichen beginnen.

?DZ FALSCH
FALSCH

Kapitel II/4

Die Prozeduren

Die in diesem Kapitel vorgestellten Grundwörter dienen dazu, Prozeduren zu definieren und zu untersuchen.

4.1 Definieren

PR *Wort Parameter1 Parameter2* (Befehl)

- dient zur Definition einer Prozedur, deren Name durch das Wort bestimmt wird und die, falls erforderlich, die genannten Parameter benutzt.

Jeder Parametername muß mit einem Doppelpunkt (:) beginnen.

Wenn der Befehl PR im Direktmodus benutzt wird, erscheint als Eingabeaufforderung das Zeichen >. Damit wird angezeigt, daß mit den folgenden Eingaben eine Prozedur definiert wird. Alle Eingaben werden in diesem Fall gespeichert und nicht sofort ausgeführt.

Beispiel für eine Definition im Direktmodus:

```
?PR VIERECK :SEITE
>WH 4 [VW :SEITE RE 90]
>ENDE
SIE DEFINIEREN GERADE VIERECK
```

Eine Prozedur kann nicht innerhalb einer anderen definiert werden. Jede Definition beginnt mit einem eigenen PR-Befehl.

```
?PR DEFINITION
>DZ [NEUE PROZEDUR EINGEBEN:]
>PR NEUPROZ
>ENDE
SIE DEFINIEREN GERADE DEFINITION
```

```
?DEFINITION
NEUE PROZEDUR EINGEBEN:
>DZ "BEGRUESSUNG
>ENDE
SIE DEFINIEREN GERADE NEUPROZ

?ZE "NEUPROZ
PR NEUPROZ
DZ "BEGRUESSUNG.
ENDE
```

ENDE

(Spezialwort)

- dient zur Beendigung einer Prozedurdefinition. ENDE ist nur als letztes Wort in einer Prozedur sinnvoll und muß in einer separaten Logo-Programmzeile stehen. Sobald Sie eine Definition mit dem Wort ENDE beenden, erscheint die Meldung:

SIE DEFINIEREN GERADE...

Anschließend befinden Sie sich wieder im Direktmodus.

Im Editiermodus (s. Kapitel II/1 1) können Sie beliebig viele Prozeduren nacheinander definieren. Auch in diesem Fall muß jede Prozedur mit dem Wort ENDE abgeschlossen werden.

Sie können eine Prozedur als Liste behandeln:

```
PR DEF :LISTE
AUSGANG 4
ELAUSGABE :LISTE
AUSGANG 1
EINGANG 4
ENDE

PR ELAUSGABE :LISTE
WENN LEER? :LISTE [RK]
DZ ERSTES :LISTE
ELAUSGABE OE :LISTE
ENDE
```

Der Parameter von DEF ist eine Liste, deren Elemente die einzelnen Zeilen der zu definierenden Prozedur sind.

```
?DEF [[PR DREIECK :X] [WH 3[VW :X RE 12! 0]] [ENDE]]
```

4.2 Prüfen

PR? Wort (Operation)
(Prozedur?) (Log)

- liefert WAHR, wenn das Wort der Name einer vom Benutzer definierten Prozedur ist, die sich im Arbeitsspeicher befindet (s. Kapitel II/8), sonst FALSCH.

PR? "VW liefert FALSCH

GW? Wort (Operation)
(Grundwort?) (Log)

- liefert WAHR, wenn das Wort eines der Logo-Grundwörter ist, und sonst FALSCH.

GW? "VW liefert WAHR

Die folgenden Prozeduren wandeln den Text einer bereits definierten Prozedur in eine Liste um:

```
PR TEXT :PROZEDUR
AUSGANG 4
ZE :PROZEDUR
AUSGANG 1
EINGANG 4
RG PRLISTE []
ENDE
```


PR PRLISTE :LISTE
SEI "LISTE MT LL :LISTE
WENN GLEICH? LETZTES :LISTE [ENDE] [RG !
:LISTE]
RG PRLISTE :LISTE
ENDE

?DZ TEXT "VIERECK
[PR VIERECK; SEITE][WH 4[VW :SEITE RE 90]][ENDE]

Kapitel II/5

Die Grundwörter zur Programmsteuerung

Die in diesem Kapitel beschriebenen Grundwörter dienen dazu, den Ablauf des Programms zu steuern.

Es handelt sich dabei um:

- Bedingungsanweisungen (bewirken die Ausführung einer bestimmten Anweisung, wenn die Bedingung erfüllt ist, und sonst die Ausführung einer anderen Anweisung)
- Wiederholungsanweisungen (bewirken die wiederholte Ausführung einer Anweisung oder Anweisungsfolge)
- Unterbrechungsanweisungen (bewirken die Unterbrechung der Ausführung einer Prozedur)
- Ausführungsanweisungen (bewirken die Ausführung einer Anweisung oder Anweisungsfolge)

WENN *Log Liste*

(Befehl)

WENN *Log Liste1 Liste2*

- führt im ersten Fall die Anweisungen in der Liste aus, wenn der logische Vergleich WAHR ergibt, und sonst nicht.
- führt im zweiten Fall die Anweisungen in Liste1 aus und ignoriert Liste2 , wenn der logische Vergleich WAHR ergibt. Wenn nicht, wird Liste1 ignoriert und Liste2 ausgeführt.

Hinweis:

Eine Anweisungsliste besteht aus einer Folge von Befehlen. Alle Parameter des WENN-Befehls müssen in einer Logo-Programmzeile Platz finden.

Beispiel für einen WENN-Befehl zur Programmunterbrechung:

```
PR MUSTER :X
WENN :X > 120 [RK]
VW :X RE 160
MUSTER :X + 1
ENDE
```

Beispiel für einen WENN-Befehl zur Addition der ersten N ganzen Zahlen:

```
PR ADD :N
WENN :N = 0 [RG 0][RG :N + ADD :N - 1]
ENDE

?DZ ADD 10
55
```

Es ist nicht schwierig, eine Prozedur WENN-T zu definieren, die wie das Grundwort WENN arbeitet, aber zusätzlich den Logo-Term liefert, der durch die Ausführung der Anweisungsliste erzeugt wird (Liste1 und Liste2 enthalten beide eine Operation bzw. einen Logo-Term).

```
PR WENN-T :BEDINGUNG :LISTE1 :LISTE2
WENN :BEDINGUNG [RG TUE :LISTE1] [RG TU!
E :LISTE2]
ENDE

?DZ WENN-T GLEICH? "L ERSTES "LILA ["JA! ]
["NEIN]
JA
```

WH n Liste (Befehl)
(Wiederhole)

– führt die Anweisungen in der Liste n-mal aus. n ist eine Zahl zwischen 0 und 65535.

Hinweis:

Eine Anweisungsliste besteht aus einer Folge von Befehlen, die in einer Logo-Programmzeile Platz finden müssen.

?WH 4 [VW 50 RE 90] – zeichnet ein Quadrat

Die folgende Prozedur erledigt die Strafarbeit für einen Schüler:

```
PR STRAFARBEIT
WH 100 [DZ [ICH DARF NICHT SCHWATZEN]]
ENDE
```

In einer Prozedur können mehrere Wiederholungen, verschachtelt sein:

```
PR VSCH :X
WH 10 * :X [WH 10 * :X []]
ENDE
```

Die folgende Prozedur wiederholt die Ausführung einer Anweisungsliste solange, bis eine Bedingung, vorgegeben als Liste, erfüllt ist:

```
PR SOLANGE-BIS :BEDINGUNG :LISTE
WENN TUE :BEDINGUNG[TUE :LISTE SOLANGE!
-BIS :BEDINGUNG :LISTE]
ENDE
```

```
?SEI "I 5
?SOLANGE-BIS [NICHT? GLEICH? :I 0] [DZ !
:I SEI "I :I - 1]
5
4
3
2
1
```

RK
(Rückkehr)

(Befehl)

- beendet die Ausführung der Prozedur, in der der RK-Befehl angetroffen wird. RK ist nur innerhalb einer Prozedur sinnvoll und ermöglicht die Definition von Befehlsprozeduren (vergl. RG und AUSSTIEG).

```
PR ZAHLEN :N
WENN :N = 5 [RK]
DZ :N
ZAHLEN :N + 1
ENDE
```

```
?ZAHLEN 1
1
2
3
4
```

RG Term (Rückgabe)

(Befehl)

- beendet die Ausführung der Prozedur, in der der RG-Befehl angetroffen wird, und gibt den angegebenen Term zurück. Der Term kann auch der Parameter der aufgerufenen Prozedur sein. RG ist nur innerhalb einer Prozedur sinnvoll und erlaubt die Definition von Operationsprozeduren (vergl. RK).

```
PR RUECKW :BUCHST
WENN LEER? :BUCHST [RG " ] [RG WORT LET!
ZTES :BUCHST RUECKW OL :BUCHST]
ENDE
```

```
?DZ RUECKW "HALLO
OLLAH
```

AUSSTIEG

(Befehl)

- unterbricht die momentane Programmausführung und setzt das System in den vorangegangenen Zustand zurück. Ausstieg ist nur innerhalb einer Prozedur sinnvoll. Der Ausstieg-Befehl hat dieselbe Wirkung wie die gleichzeitige Betätigung der Tasten CTRL] und [C].

TUE Liste

(Befehl)
oder (Operation)

- führt die Anweisungen in der Liste aus. Wenn die Liste eine Folge von Befehlen enthält, werden diese wie in einer Logo-Programmzeile abgearbeitet. TUE kann jedoch auch als Operation arbeiten. Wenn die Liste eine Operation enthält (bzw. einen Term), wird diese Operation ausgeführt, und TUE gibt das Ergebnis zurück; verhält sich also wie eine Operation.

```
?TUE [VW 50 LI 90 VW 50]
```

```
?SEI "X [SUMME 2 3]
```

```
?DZ TUE :X
```

```
5
```

```
?DZ TUE [SUMME 4 PROD 3 6]
```

```
22
```

TUE funktioniert folgendermaßen:

Sobald TUE seinen Parameter zugewiesen bekommt (der eine Liste sein kann), wird dieser anstelle von TUE in die Programmzeile eingesetzt, die Logo dann ausführt.

```
?DZ TUE [2 DZ] 3
```

```
2
```

```
3
```

Der Parameter von TUE ist in diesem Fall die Liste [2 DZ]. Sie wurde anstelle von TUE zwischen DZ und 3 eingesetzt. Ausgeführt wurde also die Zeile:

```
DZ 2 DZ 3
```

Dazu ein weiteres Beispiel:

```
?DZ PROD TUE LISTE "SUMME 2 3 4
```

```
20
```

Der Parameter von TUE, die Liste [SUMME 2], wurde zwischen PROD und 3 eingesetzt. Ausgeführt wurde also die Zeile:

```
DZ PROD SUMME 2 3 4
```

Mit Hilfe des Grundworts TUE lassen sich auf einfache Weise neue Steueranweisungen (s. die Prozeduren SOLANGE-BIS und WENN-T in diesem Kapitel) definieren.

Die runden Klammern

Eine in runden Klammern eingeschlossene Anweisungsfolge muß folgenden zwei Regeln entsprechend aufgebaut sein:

1. Die erste Anweisung muß eine Operation sein (bzw. ein Term).
2. Die folgenden Anweisungen (falls vorhanden) können Befehle sein.

Bei der Ausführung „merkt“ sich Logo den durch die Operation übergebenen (oder direkt eingegebenen) Term, während alle folgenden Befehle ausgeführt werden, bis die schließende Klammer erreicht ist. Der Term wird also durch die Klammern an die aufrufende Prozedur übergeben. Die Klammern verhalten sich also wie eine Operation: sie übergeben immer einen Term.

?DZ 2 + 3 * 4
14 (Multiplikation vor Addition)

?DZ (2 + 3) * 4
20 (die Klammern haben den Wert 5 übergeben)

?DZ (2 DZ 3)
3 (die Klammern haben erst 2 übergeben und
2 dann 3 geschrieben)

Man kann die Objekte zweier Namen austauschen, ohne einen dritten Namen zu benutzen:

```
?SEI "X 4
?SEI "Y 7
? SEI "X (:Y SEI "Y :X)
?DZ :X.
7
DZ :Y
4
```

Kapitel II/6

Die Namen

In Logo kann ein Wort der Name eines Terms sein. Der Term ist dann das „Objekt“ dieses Wortes. Dieses Objekt kann, falls es ein Wort ist, wiederum der Name eines anderen Objekts sein. In diesem Kapitel stellen wir die Grundwörter vor, die zur Verarbeitung von Namen und Objekten dienen.

SEI *Wort Term*

(Befehl)

- gibt dem Term den durch das Wort angegebenen Namen. Jedem Logo-Term, ob Wort oder Liste, kann ein Name zugewiesen werden.

?SEI "LAENDER [FRANKREICH DEUTSCHLAND E!
NGLAND]

?DZ :LAENDER
FRANKREICH DEUTSCHLAND ENGLAND

?SEI "DEUTSCHLAND "BONN

?DZ :DEUTSCHLAND
BONN

WERT *Wort*

(Operation)
(Term)

- liefert den Term, der durch das Wort bezeichnet wird. Handelt es sich bei dem Wort um einen Namen, so wird dessen Objekt übergeben. WERT kann durch den Doppelpunkt ersetzt werden.

?DZ WERT "LAENDER
FRANKREICH DEUTSCHLAND ENGLAND

?DZ :LAENDER
FRANKREICH DEUTSCHLAND ENGLAND

?DZ WERT ERSTES OE :LAENDER
BONN

Wenn Sie jedem Land eine Stadt zugeordnet haben, können Sie folgende Prozedur benutzen:

```
PR ERDKUNDE :LAENDER
WENN LEER? :LAENDER [RK]
ABFRAGE ERSTES :LAENDER
ERDKUNDE OE :LAENDER
ENDE
```

```
PR ABFRAGE :STAATEN
DZ SATZ [WIE HEISST DIE HAUPTSTADT VON !
] :STAATEN
WENN GLEICH? ERSTES LL WERT :STAATEN [DI
Z [RICHTIG]] [DZ [SO? SEIT WANN?]]
ENDE
```

Die folgende Prozedur sucht die entsprechenden Hauptstädte:

```
PR LLISTE :LAENDER
WENN LEER? :LAENDER [RK]
DZ WERT ERSTES :LAENDER
LLISTE OE :LAENDER
ENDE
```

Hinweis:

Namen können nicht nur mit dem Befehl SEI zugewiesen werden. Bei jedem Aufruf einer Prozedur, die Parameter besitzt, sind die als Parameter benutzten Terme die Objekte der Parameternamen (und werden mit vorangestellten Doppelpunkten in der Titelzeile aufgeführt).

NAME? *Wort*

(Operation)
(Log)

- liefert WAHR, wenn das Wort der Name eines Objekts ist, und sonst FALSCH.

Kapitel II/7

Arithmetik

Die Zahlen

In Logo können Sie sowohl positive als auch negative Ganz- und Dezimalzahlen verarbeiten. Logo akzeptiert Zahlen im Bereich zwischen $-9.999999E120$ und $9.999999E120$. Dabei werden nur die ersten sieben Nachkommastellen berücksichtigt (die 7. Stelle kann gerundet sein). Die in einer Programmzeile angegebenen Zahlen für arithmetische Operationen werden von Logo in Exponentenschreibweise (wissenschaftliche Notation) dargestellt, wenn ihr Absolutwert kleiner als 0.1 oder größer als $1E7$ ist. Zahlen muß in einer Programmzeile kein Doppelpunkt vorangestellt werden.

```
?DZ 12345.6789  
12345.678
```

```
?DZ 12345678910  
1.2345678E10
```

Negative Zahlen oder Exponenten werden durch ein Minuszeichen (-) unmittelbar vor dem entsprechenden Wert gekennzeichnet.

Die arithmetischen Operationen

Die Mehrzahl der arithmetischen Operationen mit zwei Parametern kann sowohl in der Normalform als auch in der Umkehrform niedergeschrieben werden.

1. Notation in Umkehrform

In der Umkehrform (auch „Umgekehrte Polnische Notation“ genannt) werden Grundwörter (und keine Symbole) für die entsprechende arithmetische Operation benutzt. Die beiden zu verknüpfenden Parameter folgen auf das entsprechende Grundwort:

?DZ SUMME 2 3
5

?DZ KLEINER? 2 3
WAHR

?DZ SUMME 2 PROD 6 DIFF 4 REST 3 2
20

Im letzten Beispiel wird zuerst REST 3 2 berechnet und das Ergebnis (1) als zweiter Parameter für DIFF benutzt. DIFF 4 1 ergibt 3, den zweiten Parameter für PROD. PROD 6 3 wiederum liefert 18 als zweiten Parameter für SUMME, und SUMME 2 18 ergibt schließlich 20. Bei dieser Schreibweise liegt die Reihenfolge der Operationen fest, wodurch Zuordnungsfehler vermieden werden.

2. Notation in Normalform

In der Normalform werden Operationssymbole benutzt. Das entsprechende Symbol steht zwischen den beiden zu verknüpfenden Zahlen, die vom Symbol durch ein Leerzeichen getrennt sein müssen. In der Normalform gilt folgende Reihenfolge: Division vor Multiplikation, Multiplikation vor Subtraktion und Addition, Subtraktion und Addition vor Vergleichen (größer, kleiner, gleich).

Die Prioritätenfolge der Symbole lautet also: / * - + < > =

?DZ 3 + 8 * 7
59

Die Reihenfolge der Ausführung kann durch Klammern geändert werden:

?DZ (3 + 8) * 7
77

Benutzen Sie in der Normalform möglichst viele Klammern, um sicherzustellen, daß die Berechnung in der gewünschten Reihenfolge ausgeführt wird!

Zahlen sind besondere Wörter, für die alle Prozeduren, die auf normale Wörter angewendet werden, ebenfalls benutzt werden können.

Die folgende Prozedur erkennt Zahlen:

```
PR GZAHL :WORT
WENN ZAHL? :WORT [DZ LETZTES :WORT] [DZ!
[KEINE ZAHL!]]
ENDE
```

Hinweis: Handelt es sich bei einem Wort um eine Zahl, so werden nur die darstellbaren Stellen berücksichtigt:

```
?DZ LETZTES 123456789
8
```

```
?DZ 123456789
1.2345678E8
```

7.1 Die arithmetischen Operationen

SUMME $n1\ n2$ (Operation)
 $n1 + n2$ (n)

liefert die Summe von $n1$ und $n2$.

```
?DZ SUMME 10.5 3.01
13.51
```

```
?DZ 25 + -3
22
```

DIFF $n1\ n2$
(Differenz)
 $n1 - n2$

(Operation)
(n)

liefert die Differenz zwischen $n2$ und $n1$.

?DZ DIFF 50 8
42

?DZ 10 - 5
5

- n

(Operation)
(n)

kehrt das Vorzeichen von n um.

?DZ - (3 - 4)
1

Hinweis: Verwechseln Sie nicht folgende Symbole:

- das Minuszeichen (-) unmittelbar vor einer Ziffer;
- das Subtraktionssymbol (-) zwischen zwei Zahlen, das in Leerzeichen eingeschlossen ist;
- die Vorzeichenumkehroperation (-), auf die eine positive oder negative Zahl folgt.

PROD $n1\ n2$
(Produkt)
 $n1 * n2$

(Operation)
(n)

liefert das Produkt von $n1$ und $n2$.

?DZ PROD 4 8
32

?VW 50 * 2

Der Igel bewegt sich 100 Schritte vorwärts.

DIV *n1 n2* (Operation)
(Division) (n)
n1 / n2

liefert das Ergebnis der Division von *n1* durch *n2*. Der zweite Parameter muß ungleich Null sein.

```
?DZ DIV 30 8  
3.75
```

QUOT *n1 n2* (Operation)
(Quotient) (n)

liefert den ganzzahligen Teil des Quotienten der Division *n1* durch *n2*. Der zweite Parameter muß ungleich Null sein.

```
?DZ QUOT 30 8  
3
```

REST *n1 n2* (Operation)
(n)

liefert den ganzzahligen Rest der Division *n1* durch *n2*. Der zweite Parameter muß ungleich Null sein.

```
?DZ REST 30 8  
6
```

Prozedur zur Berechnung einer Zahl modulo 3:

```
PR MODULODREI :X  
WENN :X  
< 0 [RG (REST :X 3) + 3] [RG RE!  
ST :X 3]  
ENDE
```

7.2 Die Vergleiche

KLEINER? $n1\ n2$

$n1 < n2$

(Operation)

(Log)

liefert WAHR, wenn $n1$ kleiner ist als $n2$, sonst FALSCH.

?DZ 3 < 8

WAHR

In einer Prozedur:

PR ZWIWERT :A :B :C

RG UND? :A < :B :B < :C

ENDE

DZ ZWIWERT 3 5 7

WAHR

?DZ ZWIWERT 5 3 7

FALSCH

GROESSER? $n1\ n2$

$n1 > n2$

(Operation)

(Log)

liefert WAHR, wenn $n1$ größer ist als $n2$, und sonst FALSCH.

?DZ 3 > 8

FALSCH

GLEICH?

=

liefert WAHR, wenn die zwei verglichenen Parameter identisch sind, und sonst FALSCH (s. Kapitel II/2).

? DZ GLEICH? 5 5

WAHR

?DZ 2 + 3 = 6

FALSCH

7.3 Weitere arithmetische Funktionen

INT n (Operation)
(Integer = Ganzzahl) (n)

liefert den ganzzahligen (mathematisch gerundeten) Teil einer Zahl.

?DZ INT 4.8989
4

?DZ INT -1.5
-2

ZZ n (Operation)
(Zufallszahl) (n)

liefert eine ganzzahlige positive Zufallszahl zwischen 0 und $(n-1)$.
 n muß gleich oder größer als 1 sein.

Die folgende Prozedur liefert eine Zufallszahl zwischen 1 und 6 (einschließlich):

```
PR ZZAHL  
RG 1 + ZZ 6  
ENDE
```

Für die zufällige Auswahl eines Elements aus einer Liste können Sie die bereits in Kapitel II/2.2 definierte Prozedur ZWAHL benutzen.

SIN n (Operation)
(Sinus) (n)

liefert den Sinus von n , wobei n die Gradangabe für einen Winkel darstellt.

?DZ SIN 45
0.707101

COS n (Operation)
(Cosinus) (n)

liefert den Cosinus von n , wobei n die Gradangabe für einen Winkel darstellt.

```
?DZ COS 45  
0.707101
```

QW n (Operation)
(Quadratwurzel) (n)

liefert die Quadratwurzel von n , wobei n gleich oder größer als Null sein muß.

```
?DZ QW 9  
3
```

Einige klassische Berechnungen:

```
PR TANGENTE :A  
RG (SIN :A) / (COS :A)  
ENDE
```

```
PR HYPOTENUSE :X :Y  
RG ( :X * :X + :Y * :Y )  
ENDE
```

Kapitel II/8

Die Verwaltung des Arbeitsspeichers

Logo unterteilt den Speicher Ihres Computers in zwei Bereiche: den Arbeitsspeicher und den Editorspeicher. Näheres über den Editorspeicher erfahren Sie in Kapitel II/11. Im Arbeitsspeicher verwaltet Logo alle definierten Prozeduren, Namen und zugehörigen Objekte.

Logo verwaltet den Arbeitsspeicher automatisch. Sobald Logo feststellt, daß zu viele Knoten (s. Operation KNOTEN) unnötig belegt sind, reorganisiert Logo den Arbeitsspeicher und stellt diese Knoten wieder zur Verfügung. Man nennt die Reorganisation im Computerjargon auch „Garbage Collection“. Damit Sie die Reorganisation auch selbst ausführen können, gibt es das entsprechende Grundwort GC.

Die in diesem Kapitel vorgestellten Grundwörter dienen zur Überprüfung oder Änderung des Arbeitsspeichers.

8.1 Prüfen

WOERTER

(Operation)
(Liste)

liefert eine Liste, deren Elemente alle Wörter sind, die der Arbeitsspeicher momentan enthält.

Die folgende Prozedur zeigt alle Wörter des Arbeitsspeichers an, die mit einem bestimmten Zeichen beginnen:

```
PR ANFZEICHEN :LISTE :ZEICHEN
WENN LEER? :LISTE [RK]
WENN GLEICH? ERSTES ERSTES :LISTE :ZEICHEN
HEN [DZ ERSTES :LISTE]
ANFZEICHEN OE :LISTE :ZEICHEN
ENDE
```

```
?ANFZEICHEN WOERTER "S
```

Es werden alle Wörter ausgegeben, die mit dem Buchstaben S beginnen.

ZGN

(Zeige Namen)

(Befehl)

zeigt alle Namen und die zugehörigen Objekte an.

```
PR NAMEN
SEI "NAME ERSTES LL
SEI "VORNAME ERSTES LL
ENDE
```

?NAMEN
MEIER
HANS

?ZGN
SEI "VORNAME "HANS
SEI "NAME "MEIER

ZGA (Befehl)
(Zeige alles)

zeigt alle Prozedurzeilen sowie alle Namen und die zugehörigen Objekte an.

ZGT (Befehl)
(Zeige Titel)

zeigt alle Prozedurtitle an.

ZE Titel (Befehl)
(Zeige)

zeigt alle Zeilen der Prozedur mit dem angegebenen Titel an.

KNOTEN (Operation)
(n)

liefert die Anzahl freier Knoten im Arbeitsspeicher. Logo berechnet den Speicherplatz im Arbeitsspeicher in Knoten. Ein Knoten enthält fünf Bytes. Wenn Sie die Operation im Anschluß an den Befehl GC (der den Arbeitsspeicher reorganisiert) ausführen, wissen Sie, wieviel Speicherplatz Ihnen noch zur Verfügung steht.

Eine Liste belegt im Arbeitsspeicher so viele Knoten, wie sie Elemente enthält. Ein Wort benötigt für jeweils zwei Zeichen einen Knoten. Wie groß der Arbeitsspeicher insgesamt ist, hängt vom Umfang der Speichererweiterung ab, über die Sie verfügen.

8.2 Ändern

VGN Name (Befehl)
(Vergiß Namen)

löscht das Wort mit dem angegebenen Namen sowie die zugeordneten Objekte aus dem Arbeitsspeicher.

```
?SEI "WERKZEUG [HAMMER ZANGE SAEGE]
```

```
?VGN "WERKZEUG
```

```
?DZ :WERKZEUG
```

```
NAME UNBEKANNT WERKZEUG
```

VGP Titel (Befehl)
(Vergiß Prozedur)

löscht die Prozedur mit dem angegebenen Titel aus dem Arbeitsspeicher. Sie können nur Prozeduren löschen, die Sie selbst definiert haben, nicht aber die „eingebauten“ Logo-Grundwörter.

Löschen Sie z.B. Ihre Prozedur VIERECK:

```
PR VIERECK
```

```
WH 4 [VW 50 RE 90]
```

```
ENDE
```

```
?VGP "VIERECK
```

```
?VIERECK
```

```
PROZEDUR UNBEKANNT VIERECK
```

.VGA (Befehl)
(Vergiß alles)

löscht alle Einträge im Arbeitsspeicher.

Dieser Befehl ist mit Vorsicht zu benutzen, denn er löscht den gesamten Arbeitsspeicher und führt zu einer Neuinitialisierung von Logo. Legen Sie dahervor Ausführung dieses Befehls Ihre DOS-Diskette in Laufwerk 0 ein, wenn Sie mit einer Diskettenlaufwerk arbeiten.

GC
(Garbage Collection)

(Befehl)

reorganisiert den Arbeitsspeicher. Diese Reorganisation findet von Zeit zu Zeit automatisch statt, wenn Logo feststellt, daß unnötig viele Knoten belegt sind. Um zu verhindern, daß Logo die (teilweise sehr zeitaufwendige) Reorganisation während der Programmausführung durchführt (und dazu das Programm unterbricht), können Sie die Reorganisation vor Ausführung eines Programms mit dem Befehl GC selbst auslösen.

Lassen Sie die Operation KNOTEN vor und nach dem Befehl GC ausführen, wenn Sie wissen wollen, wieviele Knoten durch die Reorganisation wieder frei wurden.

```
PR REORG
DZ SATZ - DIFF (KNOTEN GC) KNOTEN [BEFR!
EITE KNOTEN]
ENDE
```

```
PR KNOTENFRESSER
DZ SATZ [ES SIND NOCH] SATZ KNOTEN [KNO!
TEN FREI]
KNOTENFRESSER
ENDE
```

Kapitel II/9

Die Verbindung mit der Außenwelt

Für Logo gehört alles, was sich nicht innerhalb des Arbeitsspeichers abspielt, zur Außenwelt. Die Verbindung zur Außenwelt wird hergestellt über die Tastatur und den Bildschirm, aber auch über den Editor, die Musik, die Dateien, den Lightpen, die Joysticks, usw.

In diesem Kapitel werden alle Grundwörter vorgestellt, die zur Kommunikation mit der Außenwelt dienen (mit Ausnahme der Grundwörter zur Bearbeitung von Dateien, die in Kapitel II/10 beschrieben sind).

9.1 Die Tastatur

Einzelheiten über die Benutzung der Tastatur finden Sie im Abschnitt „Die Tastatur“ zu Beginn dieses Buches.

Die folgenden Grundwörter dienen zur Kommunikation mit der Tastatur. Sie werden in Prozeduren benutzt, um Daten während der Programmausführung einzugeben.

LZ (Operation)
(Lies Zeichen) (Z)

übergibt das nächste Zeichen, das über die Tastatur eingegeben wird, an das Programm, ohne es auf dem Bildschirm darzustellen, und ohne daß Sie die RETURN-Taste drücken müssen. Auch das Eingabezeichen (?) erscheint nicht. Nur der Cursor ist solange sichtbar, bis Sie eine Taste gedrückt haben.

Mit folgender Prozedur können Sie den Igel mit Hilfe der Tastatur über den Bildschirm bewegen:

```
PR STEUERN :T
WENN :T = "R [VW 10 RE 45]
WENN :T = "L [VW 10 LI 45]
STEUERN LZ.
ENDE
```

LL (Operation)
(Lies Liste) (Liste)

übergibt die nächste Zeile, die über die Tastatur eingegeben wird, in Form einer Liste an das Programm. Das Eingabezeichen (?) erscheint nicht, aber der Cursor ist solange sichtbar, bis Sie die Eingabe mit [RETURN] beendet haben. Alle mit LL gelesenen Zeichen werden auf dem Bildschirm dargestellt.

Hinweis: Steuerzeichen bleiben unberücksichtigt.

Die folgende Prozedur vergleicht einen einzugebenden Namen mit einer Namenliste:

```
SEI "VORNAMEN [HANS MARIANNE PETER PAUL!  
A KARL KAROLINE]
```

```
PR VERGLEICH  
DZ [WIE HEISSEN SIE MIT VORNAMEN?]  
SEI "NAME LL  
WENN EL? ERSTES :NAME :VORNAMEN [DZ [ICH!  
KENNE SIE]] [SEI "VORNAME ME ERSTE!  
S :NAME :VORNAMEN DZ [ICH KENNE SIE NIC!  
HT]]  
ENDE
```

TASTE?

(Operation)
(Log)

liefert WAHR, wenn eine Taste gedrückt ist, und FALSCH, wenn nicht.

Mit folgender Prozedur können Sie den Igel umherirren lassen und ihn durch einen Tastendruck stoppen:

```
PR IRRWEG  
VW ZZ 50 RE 30  
WENN TASTE? [RK]  
IRRWEG  
ENDE
```

9.2 Der Bildschirm

Der Bildschirm ist die visuelle Verbindung zwischen Logo und dem Benutzer. Er macht die Welt des Igels sichtbar und dient darüber hinaus zur Darstellung von Texten.

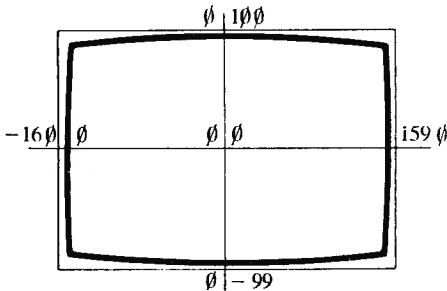
Folgende den Bildschirm betreffende Grundwörter wurden bereits im ersten Kapitel vorgestellt:

SR, SHGF, LB, LS

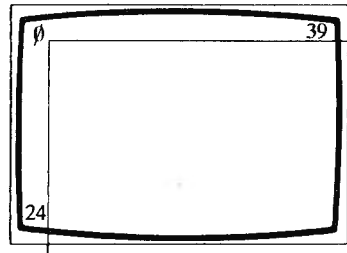
Vergleichen Sie dazu auch Anhang B.

Für den Bildschirm gibt es zwei Koordinatensysteme:

Grafikbildschirm



Textbildschirm



TB *n*

(Teile Bildschirm)

(Befehl)

teilt den Bildschirm in einen Text- und einen Grafikbereich. Der Textbereich umfaßt die unteren *n* Zeilen, wobei *n* zwischen 1 und 25 (einschließlich) liegen darf.

Mit TB 25 wird der gesamte Bildschirm zum Textbereich, jedoch wird in diesem Fall nach jedem Befehl, der den Zustand des Iglers ändert, der Textbereich in einen Grafikbereich umgewandelt.

LT

(Lösche Textbereich)

(Befehl)

löscht den Textbereich und positioniert den Cursor in die linke obere Ecke dieses Bereichs.

DZ *Term*

(Drucke Zeile *Term*)

(Befehl)

gibt den angegebenen *Term* aus und setzt den Cursor in die folgende Zeile. Listen werden ohne die eckigen Klammern ausgegeben, Wörter ohne Anführungszeichen (Vergleichen Sie dazu DRUCKE und ZEIGE).

?DZ "HALLO
HALLO

?DZ [HALLO UND GUTEN TAG]
HALLO UND GUTEN TAG

?DZ [HALLO [PETER KARL PAUL]]
HALLO [PETER KARL PAUL]

DRUCKE *Term*

(Befehl)

gibt den angegebenen Term aus. Der Cursor bleibt am Ende des Textes stehen. Listen werden ohne die eckigen Klammern ausgegeben, Wörter ohne Anführungszeichen (Vergleichen Sie dazu DZ und ZEIGE).

PR SERIE :WORT
WENN LEER? :WORT [DZ [] RK]
DRUCKE ERSTES :WORT DRUCKE [.]
SERIE OE :WORT
ENDE

?SERIE "ABCD
A.B.C.D.

PR ZEIGE :DING
DZ ME :DING []
ENDE

?DZ [KUCKUCK]
KUCKUCK
?ZEIGE [KUCKUCK]
[KUCKUCK]

SB *Liste* (Setze **B**linker)

(Liste)

setzt den Cursor an die in der Liste angegebene Position. Die Liste enthält zwei Zahlen. Die erste (0-39) ist die Bildschirmspalte, die zweite (0-24) die Bildschirmzeile.

Im Direktmodus erscheint vor dem Cursor das Eingabezeichen (?) an der angegebenen Position. Nach Drücken der ENTER-Taste springt der Cursor an den Anfang der nächsten Zeile.

SB [20 12] \

Der Cursor erscheint in der Bildschirmmitte.

Auf dem TO7-70 ist es möglich, die Cursorposition mit folgender Prozedur zu bestimmen:

```
PR CURSPOS
RG LISTE -1 + .HOLE 24608 .HOLE 24603
ENDE
```

STF *n* (Befehl)
(Setze Textfarbe)

bewirkt, daß alle folgenden Zeichen in der Farbe mit dem Farbcode *n* ausgegeben werden (die Farbcodes finden Sie in Kapitel II/1.2).

So werden z.B nach STF 5 alle folgenden Zeichen violett dargestellt.

STHF *n* (Befehl)
(Setze Texthintergrundfarbe)

bewirkt, daß alle folgenden Zeichen auf einem Hintergrund in der Farbe mit dem Farbcode *n* ausgegeben werden, d.h. in einem kleinen aus 8 mal 8 Punkten bestehenden farbigen Feld stehen (die Farbcodes finden Sie in Kapitel II/1.2).

So werden z.B. nach STHF 2 alle folgenden Zeichen in einem kleinen grünen Feld ausgegeben.

Wenn der Cursor in der letzten Bildschirmzeile steht, bewirkt die Betätigung der ENTER-Taste, daß eine neue Zeile in der momentanen Hintergrundfarbe erscheint.

Die folgenden Prozeduren ermöglichen die Zentrierung und Hervorhebung eines Titels. Sie benötigen dazu die die Prozedur ZAEHLEN aus Kapitel II/2.1).

```
PR ZENTR :WORT
SB SATZ (39 - ZAEHLEN :WORT) / 2 12
ENDE
```

```
PR TITEL :WORT :I
WENN GLEICH? :I 2 [SEI "I :I + 1]
WENN LEER? :WORT [DZ " STHF 0 RK]
STHF :I DRUCKE ERSTES :WORT
ENDE
```

```
PR BETITELN :WORT
LT
ZENTR :WORT
TITEL :WORT 1
ENDE
```

```
?BETITELN [HALLO]
```

9.3 Der Light-Pen

KONTACT? (Operation)
(Log)

liefert WAHR, wenn der Schalter an der Spitze des Light-Pens geschlossen ist, und FALSCH, wenn nicht.

LGORT (Operation)
(Light-Pen Ort) (Liste)

liefert die Koordinaten des Bildschirmpunktes, auf den der Light-Pen zeigt. LGORT gibt eine leere Liste zurück, wenn die Spitze des Light-Pens zu weit vom Bildschirm entfernt ist oder auf einen roten, einen schwarzen oder einen Randbereich des Bildschirms zeigt, oder wenn eine zu geringe Bildschirmhelligkeit eingestellt ist.

Die beiden folgenden Prozeduren dienen dazu, die Koordinaten des Light-Pens auszugeben, sobald die von LGORT gelieferte Liste nicht mehr leer ist:

```
PR POSOPT
RG SUCHOPT LGORT
ENDE

PR SUCHOPT :LISTE
WENN LEER? :LISTE [RG SUCHOPT LGORT]
RG :LISTE
ENDE
```

?DZ POSOPT

Logo wartet, bis der Light-Pen auf den Bildschirm zeigt...

Mit Hilfe der Prozedur FARBGRIFSEL können Sie eine Stiftfarbe wählen und mit dem Light-Pen zeichnen:

```
PR FARBGRIFSEL :S
WENN KONTAKT? [SA][SH SSF SF + 1]
WENN NICHT? LEER? :S [SO :S]
FARBGRIFSEL LGORT
ENDE

?FARBGRIFSEL LGORT
```

REGULIEREN

Dieser Befehl dient zum Einstellen des Light-Pens.

Folgendes Vorgehen:

1. Mit STHF die Hintergrundfarbe in jede andere Farbe als Schwarz ändern.
2. REGULIEREN eintippen und den Light-Pen solange auf den Bildschirm drücken, bis der erschienene Balken genau unter der Light-Pen-Spitze liegt.

9.4 Die Joysticks

KNOPF? *n* (Operation)
(Log)

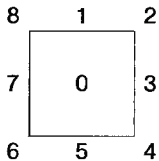
liefert WAHR, wenn der Action-Knopf des Joysticks *n* (0 oder 1) gedrückt ist, sonst FALSCH.

STEUER n

(Operation)
(n)

liefert eine Zahl zwischen 0 und 8 entsprechend der Richtung des Joysticks n (0 oder 1).

Den Richtungen sind folgende Zahlen zugeordnet:



Die folgende Prozedur ermöglicht es, den Igel mit beiden Joysticks zu lenken und die Zeichenfarbe mit dem Action-Knopf zu bestimmen:

```
PR LENKEN
SKURS 45 * STEUER 0
VW 5 * STEUER 1
WENN KNOPF? 0 [SSF SF + 1]
WENN KNOPF? 1 [SH] [SA]
LENKEN
ENDE
```

9.5 Die Musik

OKTAVE n

(Befehl)

bestimmt die Oktave n , in der alle folgenden Noten gespielt werden. n darf zwischen 1 (tief) und 5 (hoch) liegen. Standardmäßig benutzt Logo die 4. Oktave.

DAUER

(Befehl)

bestimmt die Tondauer aller folgenden Noten. n darf zwischen 1 und 96 liegen. Standardmäßig benutzt Logo Viertelnoten (Tondauer 24).

Notenwert	Tondauertabelle		
	normal	punktiert	Triole
ganz	96		64
halb	48	72	32
viertel	24	36	16
achtel	12	18	8
sechzehntel	6	9	4
zweiunddreißigstel	3		2

KLANG *n* (Befehl)

bestimmt den Klangwert *n* aller folgenden Noten. *n* darf zwischen 0 und 255 liegen. Standardmäßig benutzt Logo den Klangwert 0.

TEMPO *n* (Befehl)

bestimmt das Tempo für alle nachfolgenden Noten. *n* darf zwischen 1 und 255 liegen. Standardmäßig benutzt Logo das Tempo 5. TEMPO 1 ist langsam (largo), TEMPO 255 sehr schnell (prestissimo).

SPIELE *Wort* (Befehl)

spielt die in dem Wort angegebenen Noten. Folgende Notenbezeichnungen stehen zur Verfügung: DO, RE, MI, FA, SO (für SOL), LA, SI und PA (für Pause, eine „stille“ Note). Jede dieser Noten kann durch ein Erhöhungszeichen (#) oder ein Mollzeichen (<b) geändert werden. Beide Zeichen wirken sich nur auf die unmittelbar folgende Note aus.

?SPIELE "DOS0

?SPIELE "DOMIFA#FAFA

PR UEBUNG :N

WENN :N > 50 [RK]

TEMPO :N

DAUER 48 OKTAVE 2 SPIELE "DOREMIFASOLASI

OKTAVE 3 SPIELE "DO

DAUER 24 SPIELE "PA

UEBUNG :N + 10

ENDE

?UEBUNG 5

9.6 Die anderen Peripheriegeräte

Die folgenden Grundwörter dienen zur Kommunikation mit den restlichen Peripheriegeräten.

EINGANG n (Befehl)

bestimmt die Nummer n des Eingabekanals (Kanalcodes s. unten). Logo liest daraufhin alle folgenden Informationen und Anweisungen über den angegebenen Kanal.

Hinweis: Der Befehl EINGANG ist unnötig für die Kanäle 0 und 2.

AUSGANG n (Befehl)

bestimmt die Nummer n des Ausgabekanals. Logo gibt daraufhin alle folgenden Ausgaben über den angegebenen Kanal aus. Die Grundwörter zur Bildschirmverwaltung, insbesondere die Wörter DZ und DRUCKE, gelten auch für den neuen Kanal in gewohnter Weise.

Kanalcodes:

- 0 Pseudokanal
- 1 Tastatur
- 2 Drucker
- 3 serielle Schnittstelle
- 4 Editor

Der Kanal 0 ist ein Pseudokanal, dem kein Peripheriegerät zugeordnet ist. Mit dem Befehl AUSGANG 0 kann jede Ausgabe unterdrückt werden.

Der Kanal 1 dient zur Ausgabe auf den Bildschirm und zur Eingabe über die Tastatur. Logo benutzt diesen Kanal standardmäßig.

Der Kanal 2 dient zur Ausgabe auf den Drucker. Dieser sollte vor dem Einschalten des Computers angeschlossen und eingeschaltet werden.

Der Kanal 3 dient zur Kommunikation mit allen Peripheriegeräten, die über die serielle Schnittstelle angeschlossen sind. Vergewissern Sie sich anhand der Dokumentation, daß die Verbindung korrekt ist und Sie die richtigen Geräte angeschlossen haben.

Der Kanal 4 dient zur Ausgabe des Editorinhalts oder zur Eingabe in den Editor.

Hinweis: Sie haben die Möglichkeit, neue Ein- oder Ausgänge zu definieren. Die Eingangs- oder Ausgangskanalnummer muß dabei der Einsprungadresse einer geeigneten Assembleroutine entsprechen, die Sie in den an Logo anschließenden Adressbereich laden müssen (s. dazu die Befehle .RES und BLADE). Die ein- oder auszugebenden Zeichen werden im Register B übergeben.

Mit folgender Prozedur können Sie den Inhalt des Editors ausdrucken:

```
PR EDRUCK
EINGANG 4
AUSGANG 1
AUSGANG 2
EAUS LL
ENDE
```

```
PR EAUS :L
EAUS LL
ENDE
```

Nach der Eingabe dieser beiden Prozeduren gibt Logo den Inhalt des Editors über den Drucker aus, sobald Sie die Prozedur EDRUCK aufrufen. Nach gleichzeitiger Betätigung der Tasten [CTRL] und [C] erhalten Sie wieder das Logo-Eingabezeichen.

SZL n (Befehl)
(Setze Zeilenlänge)

legt die Länge der auszugebenden Zeilen auf n Zeichen fest. n darf zwischen 1 und 254 liegen ($0 < n < 255$). Mit Hilfe dieses Befehls können Sie eine Ausgabe an Drucker mit unterschiedlichen Druckbreiten anpassen.

KOPIE

(Befehl)

druckt den Bildschirminhalt auf dem Drucker aus.

.SER *n1 n2*

(Befehl)

(**Serielle Schnittstelle**)

bestimmt die Parameter für die serielle Schnittstelle. *n1* ist die Übertragungsrate in Baud (110, 300, 600, 1200, 2400, 4800 oder 9600), *n2* ist die Bitzahl (7 oder 8).

Kapitel II/10

Die Dateien

Dateien dienen dazu, Ihre Prozeduren und Daten aus dem Arbeitsspeicher oder dem Editorspeicher aufzubewahren, auch wenn der Computer abgeschaltet wird. Sie können die Dateien anschließend wieder laden und die Prozeduren erneut benutzen, ohne sie wieder "per Hand" eingeben zu müssen.

Dateien können auf Kasette (mit dem Programm-Rekorder) oder auf Diskette (mit einem Diskettenlaufwerk und dem zugehörigen DOS) gespeichert werden.

Alle folgenden Grundwörter, die mit einem Sternchen (*) gekennzeichnet sind (das Sternchen gehört nicht zum Grundwort), arbeiten nur, wenn DOS geladen ist.

Wenn Sie eine Datei speichern, müssen Sie ihr einen Namen zuweisen. Ein Dateiname besteht aus maximal acht Zeichen und einer Namenserweiterung, die aus einem Punkt und drei Zeichen besteht. Logo benutzt automatisch .LOG als Namenserweiterung.

In einem Dateinamen (einem Logo-Wort) dürfen folgende Zeichen nicht enthalten sein: runde Klammern, Doppelpunkte und Punkte (außer dem Punkt in der Namenserweiterung).

Logo benutzt als Speichereinheit immer das Gerät, das als "aktives Speichergerät" definiert ist. Bei angeschlossenem Programm-Rekorder ist dieser automatisch das aktive Speichergerät. Wenn das Diskettenlaufwerk angeschlossen ist, benutzt Logo standardmäßig das Laufwerk 0 als aktives Speichergerät. Um das momentane aktive Speichergerät zu wechseln, steht der Befehl **SSPEICHER** zur Verfügung.

SPEICHER (Operation)
(n)

liefert die Nummer n des momentan aktiven Speichergeräts.

```
?DZ SPEICHER  
7
```

Den Speichergeräten sind folgende Nummern zugeordnet:

Diskettenlaufwerke	0, 1, 2, 3
Programm-Rekorder	7

SSPEICHER n (Befehl)
(Setze Speichergerät)

bestimmt das Speichergerät mit der Nummer n als aktives Speichergerät. Für die Diskettenlaufwerke ist n eine der Ziffern 0, 1, 2, oder 3; für den Programm-Rekorder die Ziffer 7.

Wenn kein Diskettenlaufwerk angeschlossen ist, benutzt Logo automatisch den Programm-Rekorder als aktives Speichergerät.

BW *Dateiname Liste* (Befehl)
(Bewahre)

speichert auf dem aktiven Speichergerät alle in der Liste genannten Prozeduren und Objekte in einer Datei mit dem angegebenen Dateinamen. Vor den Namen der Objekte in der Liste muß ein Anführungszeichen stehen. Steht vor einem Listenwort ein Doppelpunkt, so nimmt Logo an, daß dieses Wort eine weitere Liste bezeichnet. Mit

```
?BW "FORMEN [QUADRAT DREIECK "DIMENSIONEN]
```

werden die Prozeduren QUADRAT und DREIECK, sowie das Objekt mit dem Namen DIMENSIONEN in der Datei FORMEN gespeichert.

```
?SEI "SPIEL [SPIELEN ANFANG "SPIEL]
?BW "DAME [:SPIEL]
```

In diesem Fall werden in der Datei DAME die in der Liste SPIEL angegebenen Prozeduren und Objekte gespeichert. Dabei handelt es sich um die Prozeduren SPIELEN und ANFANG, sowie das Objekt SPIEL.

```
?BW "WT WOERTER
```

Mit dieser Anweisung werden alle Wörter aus dem Arbeitsspeicher in einer Datei mit dem Namen WT gespeichert.

LADE *Dateiname* (Befehl)

lädt vom aktiven Speichergerät den Inhalt der Datei mit dem angegebenen Dateinamen. Sobald Logo eine Prozedur geladen hat, erscheint auf dem Bildschirm die Meldung „SIE DEFINIEREN GERADE (Prozedurname)“

Falls Ihre Datei Objekte enthält, die Logo nicht zuordnen kann, wird eine entsprechende Fehlermeldung ausgegeben.

Hinweis: Sollten in einer Datei, die gerade geladen wird, eines der Grundwörter LADE, BW, EDBW, EDLADE oder ED enthalten sein, so wird dieses nicht sofort ausgeführt.

BLADE *Dateiname n* (Befehl)
(Binärdatei laden)

lädt die angegebene Binärdatei. n ist der Versatz (Offset) für die in der Datei angegebene Startadresse. Bei $n=0$ speichert Logo die Binärdaten ab der Startadresse.

Dieser Befehl sollte nur von erfahrenen Programmierern benutzt werden.

EDBW *Dateiname* (Befehl)
(Editorspeicher **bewahren**)

speichert auf dem aktiven Speichergerät den Editorinhalt unter dem angegebenen Dateinamen.

?EDBW "BRIEF.ABC

EDLADE *Dateiname* (Befehl)
(Datei in Editorspeicher **laden**)

lädt den Inhalt der angegebenen Datei (ohne ihn auszuführen) vom aktiven Speichergerät in den Editorspeicher. Um die geladenen Prozeduren auszuführen, müssen Sie zuerst den Editor aufrufen und anschließend gleichzeitig die Tasten [CTRL] und [Q] drücken. Einzelheiten dazu finden Sie in Kapitel II/11.

Hinweis: Bei zu großen Dateien besteht die Möglichkeit, daß sie nicht in den Editorspeicher passen.

Wenn Sie einen Programm-Rekorder benutzen und die Dateinamen auf der Kassette vergessen haben, können Sie mit folgender Anweisung die erste Datei auf der Kassette laden:

LADE WORT ZEICHEN 32 WORT ZEICHEN 46 ZEICHEN 32

* **FORMAT** *n* (Befehl)

formatiert die Diskette, die sich im Laufwerk *n* (0, 1, 2 oder 3) befindet. Das Logo-Diskettenformat ist kompatibel mit dem BASIC-Diskettenformat des TO7-70 und des MO5 E. Sie können daher BASIC- und Logo-Dateien auf derselben Diskette speichern. Die Formatierung einer Diskette dauert ungefähr drei Minuten.

?FORMAT 1

Die Diskette in Laufwerk 1 wird formatiert.

Hinweis: Durch die Formatierung werden alle Daten gelöscht, die sich auf der Diskette befinden.

* INHALT

(Befehl)

zeigt das Inhaltsverzeichnis der Diskette im aktiven Speichergerät an, d.h., alle Dateinamen mit Angabe des Dateiumfangs und der Dateiart.

* VERGISSDATEI *Dateiname*

(Befehl)

löscht die angegebene Datei von der Diskette aus dem aktiven Speichergerät. Dieser Befehl gilt nur für Diskettendateien und ist nicht rückgängig zu machen.

Kapitel II/11

Die Welt des Editors

Logo besitzt einen teilweise interaktiven Texteditor, mit dem Prozeduren, aber auch andere Texte erfaßt, geändert und gedruckt werden können. Der Editor ist seitenorientiert, d.h., Sie können jeweils eine ganze Bildschirmseite bearbeiten, indem Sie einfach den Cursor an die gewünschte Stelle bewegen.

Der Editor kann unabhängig von Logo benutzt werden. Der Editorinhalt wird in diesem Fall von Logo nicht als Programmtext interpretiert. Dadurch lassen sich beliebige Texte bearbeiten.

11.1 Aufrufen und Verlassen des Editors

a) Aufrufen des Editors

Sobald Sie den Editor aufgerufen haben und sich im Editiermodus befinden, färbt sich der Bildschirm blau und der Bildschirmrand sowie der Text hellblau (Einzelheiten zur Änderung dieser Farben finden Sie in Anhang B). Gleichzeitig verschwindet das Eingabezeichen (?), und nur der Cursor ist noch vorhanden.

ED
(Editor)

(Befehl)

ruft den Editor auf, ohne den Inhalt des Editorspeichers zu ändern. Der Editor ist in dem Zustand, in dem Sie ihn verlassen haben.

ED Liste

(Befehl)

ruft den Editor auf und überträgt die in der Liste angegebenen Prozeduren und Objekte zur Bearbeitung in den Editorspeicher. Vor den Namen der Objekte in der Liste muß ein Anführungszeichen stehen. Steht vor einem Listenwort ein Doppelpunkt, so nimmt Logo an, daß dieses Wort eine weitere Liste bezeichnet (wie beim Grundwort BW).

?ED [QUADRAT "SEITE]

Der Editor wird aufgerufen und die Prozedur QUADRAT sowie das Objekt SEITE in den Editorspeicher geladen.

ED []

(Befehl)

ruft den Editor auf und leert den Editorspeicher.

b) Verlassen des Editors

Es gibt zwei Möglichkeiten, den Editor zu verlassen:

mit [CTRL][Q], sodaß der Inhalt des Editorspeichers als Logo-Programmtext interpretiert wird;

mit [CTRL][C], ohne daß der Inhalt des Editorspeichers als Logo-Programmtext interpretiert wird.

[CTRL][Q]

Drücken Sie die Taste [CTRL] und drücken Sie gleichzeitig die Taste [Q]. Damit bewirken Sie, daß Logo den Inhalt des Editorspeichers interpretiert. Für jede Prozedur, die Logo übernimmt, erscheint die Meldung:

SIE DEFINIEREN GERADE...

Wenn der Editorspeicher Logo-Anweisungen enthält, werden diese normal ausgeführt. Findet Logo dagegen einen Text, der nicht als Programmzeile erkannt wird, erscheint eine Fehlermeldung, z.B.:

WAS SOLL GESCHEHEN MIT...

[CTRL][C]

Halten Sie die Taste [CTRL] nieder und drücken Sie zusätzlich die Taste [C]. Dadurch verlassen Sie den Editor, ohne daß Logo versucht, den Inhalt des Editorspeichers zu interpretieren. Verlassen Sie den Editor auf diese Weise, wenn Sie einen Text bearbeitet haben, der kein Programmtext ist.

11.2 Die Arbeit mit dem Editor

Alle Texte, die Sie mit dem Editor bearbeiten, befinden sich in einem speziellen Bereich des Arbeitsspeichers, dem Editorspeicher. Der Editorspeicher umfaßt 1000 Zeichen, die auf mehreren Bildschirmseiten dargestellt werden können.

Im Editiermodus gibt es keine Längenbegrenzung für Textzeilen. Nach jeweils 39 Zeichen erscheint automatisch ein Ausrufungszeichen (!) am Ende der Zeile, um anzudeuten, daß der Text auf der folgenden Zeile fortgesetzt wird. Logo ist in der Lage, alle Editorzeilen als Programmzeilen zu interpretieren (nach [CTRL][Q]), wenn sie kein Wort enthalten, das länger ist als 248 Zeichen.

Im Editiermodus ist es möglich, Prozeduren zu kommentieren, denn Logo läßt alle Zeichen außer acht, die sich hinter einer schließenden eckigen Klammer (]) befinden, wenn zu dieser keine entsprechende öffnende eckige Klammer ([) vorhanden ist. Alle Kommentare werden mitgespeichert.

Beim Ausdrucken werden Zeichen aus dem Editorspeicher, die nicht darstellbar sind, durch ein Sonderzeichen (den „Klammeraffen“) ersetzt.

Sobald der Editorspeicher gefüllt ist, färbt sich der Bildschirmrand rot, und es werden keine weiteren Zeichen mehr angenommen. Stattdessen ertönt bei jedem Tastendruck ein Warnton. Sie müssen dann den Editor verlassen.

Sie können den Cursor im Editiermodus mit Hilfe der Richtungstasten an jede beliebige Bildschirmposition bewegen und dort Zeichen eingeben, einfügen, kopieren oder löschen.

a) Positionieren des Cursors

Der Cursor kann mit Hilfe der Richtungstasten auf dem Bildschirm positioniert werden.

[<-] bewegt den Cursor nach links. Wenn er sich am Anfang einer Zeile befindet, bewegt er sich an das Ende der darüber liegenden Zeile (falls diese vorhanden ist).

[->] bewegt den Cursor nach rechts. Wenn er sich am Ende einer Zeile befindet, bewegt er sich an den Anfang der darunter liegenden Zeile (falls diese vorhanden ist).

[↓] bewegt den Cursor (in derselben Spalte) nach unten.

[↑] bewegt den Cursor (in derselben Spalte) nach oben.

Eine zeilenweise Positionierung ist mit Hilfe der Taste [INS] möglich:

[INS][→] setzt den Cursor an den Zeilenanfang

[INS][←] setzt den Cursor an das Zeilenende

[INS][↑] setzt den Cursor an den Textanfang

[INS][↓] setzt den Cursor an das Textende

Drücken Sie die INS-Taste einfach ein zweites Mal, wenn Sie einen vorangegangenen (möglicherweise zufälligen) INS-Tastendruck aufheben wollen.

b) Text einfügen, löschen, kopieren

Der Editor arbeitet im Einfügemodus, d.h., jedes eingegebene Zeichen belegt den Platz über dem Cursor und schiebt den vorhandenen Text ab dieser Position um ein Zeichen nach rechts.

Nach einer senkrechten Bewegung des Cursors kann es erforderlich sein, vor dem Einfügen von Text zuerst die Taste (→) zu drücken.

[ENTER] - beendet eine Zeile und beginnt eine neue (an der Cursorposition wird ein unsichtbares Wagenrücklaufzeichen eingefügt). Mit der ENTER-Taste kann eine neue Zeile begonnen oder eine vorhandene Zeile abgebrochen und am darunterliegenden Zeilenanfang fortgesetzt werden.

[DEL] - löscht das Zeichen über dem Cursor. Steht der Cursor am Zeilenende, so wird das Wagenrücklaufzeichen gelöscht, wodurch die Folgezeile an das Zeilenende herangezogen wird.

[CTRL][D] - löscht das Zeichen unmittelbar links vom Cursor.

[CTRL][S] - löscht alle Zeichen rechts vom Cursor bis zum Zeilenende und speichert die gelöschten Zeichen in einem Zwischenspeicher. Dieser ist Teil des Editorspeichers; die Zwischenspeicherung ist also nur möglich, wenn der Editorspeicher noch genügend Platz enthält. Es wird immer nur die letzte mit [CTRL][S] gelöschte Zeichenfolge gespeichert.

[CTRL][R] - fügt den Inhalt des Zwischenspeichers (also alle Zeichen, die zuletzt mit [CTRL][S] gelöscht wurden) ab der Cursorposition ein.

c) Bildlauf

Nachdem der Cursor am Ende der letzten Bildschirmzeile angekommen ist, wird der gesamte Bildschirminhalt um eine Zeile nach oben geschoben, und es erscheint eine neue Zeile am unteren Bildschirmrand, der Cursor steht am Zeilenanfang.

Kapitel II/12

Grundwörter für den erfahrenen Programmierer

.RES *a* (Befehl)
(Reserviere)

initialisiert den Arbeitsspeicher vollständig neu und sperrt ihn ab der angegebenen Adresse *a* für Logo. Dadurch steht Ihnen ein geschützter Speicherbereich für Ihre Maschinenprogramme zur Verfügung (s. dazu die Befehle BLADE, .LEGE und .RUFÉ). Hinweis: Denken Sie daran, daß auch das DOS Speicherplatz benötigt!

.HOLE *a* (Operation)
(*n*)

liefert den Inhalt des Bytes mit der Adresse *a* (eine Ganzzahl zwischen 0 und 256).

.LEGE *a x* (Befehl)

speichert den Wert *x* in dem Byte mit der Adresse *a*. *x* muß eine positive Ganzzahl sein. Ist sie größer als 256, so wird sie modulo 256 benutzt.

Hinweis: Benutzen Sie .LEGE äußerst vorsichtig, da Sie mit diesem Befehl auch Logo- oder DOS-Adressen überschreiben können.

.RUFÉ *a* (Befehl)

ruft die Maschinenroutine mit der Startadresse *a* auf und führt sie aus (geladen werden Maschinenroutinen mit dem Befehl BLADE).

Anhang A

Die Logo-Meldungen

PROZEDUR UNBEKANNT...

besagt, daß Logo die angegebene Prozedur nicht kennt. Sie haben entweder den Prozedurnamen falsch geschrieben oder die Prozedur noch nicht definiert.

FEHLENDE EINGABEN FUER...

besagt, daß Sie beim Aufruf der genannten Prozedur nicht genügend Parameter angegeben haben. Die Anzahl einzugebender Parameter wurde bei der Definition der Prozedur festgelegt.

WAS SOLL GESCHEHEN MIT...

besagt, daß Logo nicht weiß, welcher Prozedur der angegebene Parameter zuzuordnen ist. Überprüfen Sie, ob Sie zu viele Parameter eingegeben haben.

NAME UNBEKANNT...

besagt, daß Logo kein Objekt mit dem angegebenen Namen findet. Sie haben den Namen entweder falsch geschrieben oder noch nicht vergeben.

IGEL IM AUS

besagt, daß der letzte Bewegungsbefehl nicht weiter ausgeführt werden kann, da das Igelfeld auf die Bildschirmfläche begrenzt ist (RAND-Befehl) und der Igel den Bildschirmrand erreicht hat. Heben Sie die Feldbegrenzung auf oder geben Sie eine kürzere Wegstrecke an.

...MAG NICHT...

besagt, daß Sie dem angegebenen Grundwort einen Parameter der falschen Art zugewiesen haben. Überprüfen und ändern Sie den angegebenen Parameter.

SIE DEFINIEREN GERADE...

besagt, daß Logo die angegebene Prozedur in seinen Grundwortschatz aufgenommen hat.

...BEREITS VORHANDEN

besagt, daß die angegebene Prozedur bereits vorhanden ist und nicht erneut definiert werden kann. Benutzen Sie den Editor, wenn Sie die genannte Prozedur ändern wollen.

UNTERBRECHUNG !

besagt, daß das momentane Programm unterbrochen wurde und Sie sich wieder im Eingabemodus befinden. Die Unterbrechung kann durch Logo oder durch Betätigung der Tasten [CTRL][C] bewirkt worden sein.

...IN...

besagt, daß Logo in der angegebenen Prozedur einen Fehler gefunden hat. Korrigieren Sie die Prozedur mit dem Editor.

KEIN PLATZ MEHR

besagt, daß nicht mehr genügend Platz im Arbeitsspeicher vorhanden ist. Logo unterbricht ein eventuell laufendes Programm und reorganisiert den Arbeitsspeicher (Garbage Collection). Dabei werden unnütz belegte Knoten wieder freigegeben.

ZU GROSSE ZAHL

besagt, daß eine Berechnung einen Wert geliefert hat, der zu groß ist, um von Logo verarbeitet zu werden. Prüfen Sie Ihre Formel und die verwendeten Zahlen.

WORT ZU LANG

besagt, daß Logo im Editor oder in einer Datei ein Wort mit mehr als 248 Zeichen gefunden hat. Kürzen oder teilen Sie das entsprechende Wort.

IN EDITOR KEIN PLATZ MEHR

besagt, daß der Editorspeicher voll ist. Er kann nicht mehr als 1000 Zeichen aufnehmen. Verlassen Sie den Editor. Sie können den Inhalt des Editorspeichers auf Diskette oder Kassette speichern (s. Befehl EDBW).

FEHLENDE EINGABEN FUER EL

besagt, daß die übergebene Liste zu wenige Elemente enthält. Vergewissern Sie sich, daß es sich beim ersten Listenelement um eine Zahl handelt, die kleiner ist als die Anzahl der Listenelemente.

LESEN NICHT MOEGLICH

besagt, daß Logo eine Datei auf dem aktiven Speichergerät nicht lesen kann. Vergewissern Sie sich, daß alle Verbindungen hergestellt sind und die Datei sich auf dem aktiven Speichergerät befindet.

SCHREIBEN NICHT MOEGLICH

besagt, daß Logo eine Datei auf dem aktiven Speichergerät nicht speichern kann. Vergewissern Sie sich, daß alle Verbindungen hergestellt sind. Prüfen Sie bei Benutzung einer Diskettenstation, ob DOS geladen ist, ob die Diskette schreibgeschützt ist oder ob sich bereits eine Datei gleichen Namens auf der Diskette befindet.

FLOPPY NICHT BEREIT

besagt, daß der Zugriff auf die Diskette nicht ordnungsgemäß durchgeführt werden konnte. Vergewissern Sie sich, daß die Diskettenstation in Ordnung, eingeschaltet und richtig mit dem Computer verbunden ist.

Anhang B

Die ASCII-Codes und die Benutzung der Sonderzeichen

Dezimal- code

00	NULL		
01			
02	STX	[STOP]-Taste	
03			
04			
05			
06			
07		Signalton	
08	BS	[←]-Taste	
09	HT	[→]-Taste	
10	LF	[↓]-Taste Zeilenvorschub	
11	VT	[↑]-Taste	
12	FF	[CLS]-Taste	
13	CR	[ENTER]-Taste	
14	SO	Semigrafikmodus ein	
15	SI	Textmodus ein	
16			
17	DC1	Cursorblinken ein	
18	DC2	Zeichenwiederholung ein	
19			
20	DC4	Cursorblinken aus (Cursor unsichtbar)	
21			
22	SS2	[ACC]-Taste	
23			
24	CAN	Löscht ab Cursorposition bis Zeilenende	
25			
26			
27	ESC	Steuercodesequenz	
28	FS	[INS]-Taste	
29	GS	[DEL]-Taste	
30	RS	Cursor auf Seitenanfang : home	
31	US	Steuercodesequenz	
32		Leerstelle	80 P
33		!	81 Q

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

”

\$
%
&
'
(
)
*
+
,
-
.
/
0
1
2
3
4
5
6
7
8
9
:
;
<
=
>
?
@
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

R
S
T
U
V
W
X
Y
Z
[
\
]
↑
□
▣
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
{
▣
}
▣

Die Benutzung der Sonderzeichen

Einige der Sonderzeichen aus dem Codebereich 0 bis 31 dienen zur Ausführung bestimmter Sonderfunktionen auf Ihrem Computer.

Code Wirkung

- 07 gibt den Signalton aus
- 12 löscht den Bildschirm
- 14 schaltet in den Semigrafikmodus um (nur MO5 E)
- 15 schaltet in den Textmodus zurück (nur MO5 E)
- 17 macht den Cursor sichtbar (Blinken)
- 20 macht den Cursor unsichtbar
- 27 Escape-Zeichen

Das Escape-Zeichen dient zur Einleitung einer Steuercodesequenz. Der folgenden Tabelle können Sie entnehmen, welches Steuerzeichen auf das Escape-Zeichen folgen kann und welche Funktion dadurch ausgeführt wird.

MO5 E	TO7-70	Wirkung
112	76	Zeichen in Normalgröße
113	77	Zeichen in doppelter Höhe
114	78	Zeichen in doppelter Breite
115	79	Zeichen in doppelter Größe
116	105	Zeichen mit Farbe
117	104	Zeichen ohne Farbe
118	108	Mischbild aus
119	109	Mischbild an
120	106	Normaler Seitenvorschub des Bildschirms
121	110	Langsamer Seitenvorschub des Bildschirms
122	107	Seitenmodus
123	92	Inverses Bild
-	88	Zeichen verdeckt (Zeichen und Hintergrund schwarz)
-	95	Zeichen unverdeckt

Mit Hilfe der Steuerzeichensequenz **Escape-Zeichen (27)**, **Doppelkreuz (35)**, **Leerzeichen (32)** und einem Code aus der folgenden Tabelle können Sie die Text- und Hintergrundfarbe festlegen. Wird das Leerzeichen weggelassen, ändert sich nur die Zeichenfarbe.

MO5 E

	schwarz	...	orange
Text	64	...	79
Hintergrund	80	...	95

TO7-70

	schwarz	...	weiß	grau	...	orange
Text	64	...	71	112	...	119
Hintergrund	80	...	87	120	...	127

Ebenso ist es möglich, die Editorfarben mit einer Escape-Sequenz zu ändern. Beispiel:

```
SEI "ED WORT ZEICHEN 27 WORT ZEICHEN 32 ZEICHEN 85
```

Frei definierbare Zeichen (Benutzerzeichen)

Die Zeichen mit den ASCII-Codes 128 bis 255 sind frei definierbar, d.h., ihr Aussehen kann vom Benutzer mit Hilfe der Punktbefehle `.RES` und `.LEGE` beliebig festgelegt werden.

Die Adressen 24621 und 24622 (beim TO7-70) bzw. 8304 und 8305 (beim MO5 E) enthalten die beiden Bytes der Anfangsadresse der Benutzerzeichentabelle. Jedes Benutzerzeichen besteht aus 8 aufeinanderfolgenden Bytes, deren Bits das 8*8-Punktfeld des Benutzerzeichens repräsentieren. Das erste Byte definiert die untere Punktzeile des Zeichens, das zweite Byte die zweite Punktzeile von unten, usw. Für jedes Bit mit dem Wert 1 wird ein Punkt dargestellt.

Die Codes 243 bis 255 definieren standardmäßig die Umlaute und Zeichen mit Akzent. Auch sie können geändert werden.

Um einen Speicherbereich für die Benutzerzeichen zu reservieren, müssen Sie den Befehl `.RES` benutzen. Beachten Sie, daß Logo dadurch neu initialisiert wird! Legen Sie die Benutzerzeichentabelle am Ende des Speichers an, hinter den Logo-Arbeitsspeicher.

Anhang C

Automatisches Laden von Prozeduren

Wenn Sie einen TO7-70 besitzen und ein Diskettenlaufwerk oder der Programm-Rekorder angeschlossen ist, können Sie nach dem Einschalten die Option 2 (gespeichertes Programm) wählen, um Logo zusammen mit Ihren eigenen Prozeduren automatisch zu starten.

Der Computer lädt nach Wahl der Option 2 die erste Datei von der Kassette oder die Datei ABFAHRT.LOG von der DOS-Diskette in Laufwerk 0. Wenn diese Datei nicht vorhanden ist, wird eine entsprechende Meldung ausgegeben. Diese Datei können Sie benutzen, um Prozeduren oder Worte, die Sie ständig benötigen, zusammen mit Logo zu laden.

Auf dem MO5 E steht diese Möglichkeit in abgewandelter Form ebenfalls zur Verfügung, allerdings nur für das Diskettenlaufwerk. Beim Laden des DOS, d.h., bei Ausführung des Befehls .VGA, sucht Logo die Datei ABFAHRT.LOG. Wenn sie vorhanden ist, wird sie geladen, wenn nicht, arbeitet Logo ohne entsprechende Meldung weiter.

Anhang D

Endliche und unendliche Prozeduren

In Logo gibt es endliche und unendliche Prozeduren. Eine Prozedur ist unendlich, wenn sie nach ihrem Aufruf von selbst nicht wieder abbricht. Eine unendliche Prozedur ist z.B. die folgende:

```
PR ZAEHLEN :N
DZ :N
RG ZAEHLEN :N + 1
ENDE
```

Dagegen ist folgende Prozedur endlich:

```
PR UMSETZEN :L
WENN LEER? :L [RG []]
RG MT ERSTES :L UMSETZEN OE :L
ENDE
```

Unendliche Prozeduren werden von Logo iterativ abgearbeitet, d.h., immer wieder aufgerufen (auch wenn Sie WENN- oder TUE-Anweisungen enthalten) und führen daher niemals zu einem Überlauf des Stapelspeichers.

Beispiel: Erstellung einer Liste der Ganzzahlen von 1 bis N

```
PR GANZZAHL1 :N
WENN :N = 0 [RG []]
RG ME :N GANZZAHL1 :N - 1
ENDE
```

```
PR GANZZAHL2 :N
RG GANZR :N []
ENDE
```

```
PR GANZR :N :R
WENN :N = 0 [RG :R]
RG GANZR :N - 1 :N :R
ENDE
```

Vergleichen Sie GANZZAHL1 und GANZZAHL2 anhand der Werte 10, 100, 500 und 1000 für N.

Anhang E

Verzeichnis der Befehle und Operationen

Kapitel II/1: Die Welt des Igels

Die Bewegungen

VORWAERTS	VW (vorwärts)
RUECKWAERTS	RW (rückwärts)
RE	(rechts)
LI	(links)
ORT	
SO	(setze Ort)
KURS	
SKURS	(setze Kurs)

Das Igelfeld

HF	(Hintergrundfarbe)
SHGF	(setze Hintergrundfarbe)
SR	(setze Randfarbe)
VI	(verstecke Igel)
ZI	(zeige Igel)
SICHTBAR?	
LS	(lösche Schirm)
LB	(lösche Bild)
MITTE	
MAB	
SMAB	(setze Mapstab)
RAND	
RS	(Rückkehr auf gleicher Spur)
FEN	(Fenster)
PUNKT	

Der Zeichenstift

SA	(Stift ab)
SH	(Stift hoch)
SA	(Stift ab?)
SF	(Stiftfarbe)
SSF	(setze Stiftfarbe)

Kapitel II/2: Die Welt der Wörter und der Listen

Prüfen

GLEICH?

LEER?

LISTE?

WORT?

EL? (Element?)

ZAHL?

ASCII

ZEICHEN

LAENGE

Abbauen

ERSTES

OE (ohne Erstes)

LETZTES

OL (ohne Letztes)

EL (Element)

Aufbauen

WORT

SATZ

LISTE

ME (mit Erstem)

MT (mit Letztem)

Kapitel II/3: Die logischen Grundoperationen

UND?

ODER?

NICHT?

WAHR

FALSCH

Kapitel II/4: Die Prozeduren

Definieren

PR (Prozedur)
ENDE

Untersuchen

PR? (Prozedur?)
GW? (Grundwort?)

Kapitel II/5: Die Grundwörter zur Programmsteuerung

WENN
WH (wiederhole)
RK (Rückkehr)
RG (Rückgabe)
AUSSTIEG
TUE
Die runden Klammern

Kapitel II/6: Die Namen

SEI
WERT
NAME?

Kapitel II/7: Arithmetik

Die arithmetischen Operationen

SUMME
DIFF (Differenz)
PROD (Produkt)
DIV (Division)
QUOT (Quotient)
REST

Die Vergleiche

KLEINER?
GROESSER?
GLEICH?

Weitere arithmetische Funktionen

INT (Integer = Ganzzahl)
ZZ (Zufallszahl)
SIN
COS
QW (Quadratwurzel)

Kapitel II/8: Die Verwaltung des Arbeitsspeichers

Überprüfen

WOERTER
ZGN (zeige Namen)
ZGA (zeige alles)
ZGT (zeige Titel)
ZE (zeige)

Ändern

VGN (vergiß Namen)
VGP (vergiß Prozedur)
.VGA (vergiß alles)
GC (garbage collection =
Speicher reorganisieren)

Kapitel II/9: Die Verbindung mit der Außenwelt

Die Tastatur

LZ (lies Zeichen)
LL (lies Liste)
TASTE?

Der Bildschirm

TB	(teile Bildschirm)
LT	(lösche Textbereich)
DRUCKEZEILE	DZ (drucke Zeile)
DRUCKE	
SB	(setze Blinker)
STF	(setze Textfarbe)
STHF	(setze Texthintergrundfarbe)

Der Light-Pen

KONTACT?	
LGORT	(Light-Pen-Ort)
REGLE	

Die Joysticks

KNOPF?	
STEUER	

Die Musik

OKTAVE	
DAUER	
KLANG	
TEMPO	
SPIELE	

Die anderen Peripheriegeräte

EINGANG	
AUSGANG	
SZL	(setze Zeilenlänge)
KOPIE	
.SER	(serielle Schnittstelle) (nur TO7-70)

Kapitel II/10: Die Dateien

SPEICHER	(Speichergerätenummer)
SSPEICHER	(setze Speichergerät)
BW	(bewahre)

LADE
 BLADE (Binärdatei laden)
 EDBW (Editorspeicher bewahren)
 EDLADE (Datei in Editorspeicher laden)
 FORMAT
 INHALT
 VERGISSDATEI (Datei löschen)

Kapitel II/11: Die Welt des Editors

ED (Editor aufrufen)

CNT	Q
CNT	C
←	
→	
↑	
↓	

ENTREE

ED	
CNT	D
CNT	S
CNT	R
INS	←
INS	→
INS	↑
INS	↓
INS	INS

Kapitel II/12: Grundwörter für den erfahrenen Benutzer

.RES (reserviere)
 .HOLE
 .LEGE
 .RUF

Anhang F

Alphabetisches Grundwörterverzeichnis

A		G	
ASCII	51	GC	84
AUSGANG	94	GLEICH?	49 78
AUSSTIEG	68	GROESSER?	78
		GW?	63
B		H	
BLADE	98	HF	37
BW	97		
C		I	
COS	80	INHALT	100
		INT	79
D		K	
DAUER	92	KLANG	93
DIFF	76	KLEINER?	78
DIV	77	KNOPF?	91
DRUCKE	88	KNOTEN	82
DRUCKEZEILE (DZ)	87	KONTACT?	90
DZ	87	KOPIE	96
		KURS	35
E		L	
ED	101	LADE	98
EDBW	99	LAENGE	52
EDLADE	99	LB	41
EINGANG	94	LEER?	49
EL	54	LETZTES	54
EL?	50	LGORT	90
ENDE	62	LI	33
ERSTES	53	LISTE	56
		LISTE?	50
F		LL	85
FALSCH	60	LS	40
FEN	43	LT	87
FORMAT	99	LZ	85

M	
MAB	42
ME	57
MITTE	41
MT	57

N	
NAME?	72
NICHT?	60

O	
OE	53
OKTAVE	92
OL	54
ORT	34

P	
PR	61
PR?	63
PROD	76
PUNKT	44

Q	
QUOT	77
QW	80

R	
RAND	43
RE	33
REGULIEREN	91
REST	77
RG	68
RK	67
RS	43
RUECKWAERTS (RW)	32
RW (RUECKWAERTS)	32

S	
SA	45
SA?	45
SATZ	56
SB	88
SEI	71
SF	45
SH	45
SHGF	38
SICHTBAR?	40
SIN	79
SKURS	36
SMAB	42
SO	35
SPEICHER	97
SPIELE	93
SR	38
SSF	45
SSPEICHER	97
STEUER	92
STF	89
STHF	89
SUMME	75
SZL	95

T	
TASTE?	86
TB	87
TEMPO	93
TUE	69

U	
UND	59

V	
VERGISSDATEI	100
VGN	83
VGP	93
VI	39
VORWAERTS (VW)	32
VW (VORWAERTS)	32

W

WAHR	60
WENN	65
WERT	71
WH	66
WOERTER	81
WORT	55
WORT?	50

Z

ZAHL?51	
ZE	82
ZEICHEN	52
ZGA	82
ZGN	81
ZGT	82
ZI	39
ZZ	72

Punktbefehle

.HOLE	105
.LEGE	105
.RES	105
.RUF	105
.SER	96
.VGA	83

Anhang G

Verzeichnis der definierten Prozeduren

Teil I

GRUSS	15
HALLO	16
GRUSS1	18
NULL-ODER-EINS	21
HALLOHALLO	22
GRUSS2	22
SCHREIBE-INHALT	23

Teil II

1. Die Welt des Igels

RSEGMENT	33
LSEGMENT	33
KREIS	34
XKO	34
YKO	34
VIERECK	35
HOR	35
VERT	35
RANDFARBE	38
WAHL	38
ELLIPSEN	43
ABPRALL	44
AUSSEN	44
SPIRALPUNKTE	44
RADIEREN	46
RADIERHALT	46
MUSTER	47

2. Die Welt der Wörter und der Listen

WORTELEMENT	50
VERGLEICH	51
WORTCODE	51
SATZCODE	52
ALPHABET	52
ZAEHLEN	53
UMKEHR	54
ZWAHL	55
POS1	55
EINSETZEN	56

3. Die logischen Grundoperationen

NACH-INNEN	59
------------	----

4. Die Prozeduren

VIERECK	61
DEFINITION	61
NEUPROZ	62
DEF	62
DREIECK	63
ELAUSGABE	62
TEXT	63
PRLISTE	64

5. Die Grundwörter zur Programmsteuerung

MUSTER	66
ADD	66
WENN-T	66
STRAFARBEIT	67
VSCH	67
SOLANGE-BIS	67
ZAHLEN	68
RUECKW	68

6. Die Namen

ERDKUNDE	72
ABFRAGE	72
LLISTE	72

7. Arithmetik

GZAHL	75
MODULODREI	77
ZWIWERT	78
ZZAHL	79
TANGENTE	80
HYPOTENUSE	80

8. Die Verwaltung des Arbeitsspeichers

ANFZEICHEN	81
NAMEN	81
VIERECK	83
REORG	84
KNOTENFRESSER	84

9. Die Verbindung mit der Außenwelt

STEUERN	85
VERGLEICH	86
IRRWEG	86
SERIE	88
ZEIGE	88
CURSPOS	89
ZENTR	90
TITEL	90
BETITELN	90
POSOPT	91
SUCHOPT	91
FARBGRIFFEL	91
LENKEN	92
UEBUNG	93
EDRUCK	95
EAUS	95

Anhang D

ZAEHLEN	114
UMSETZEN	114
GANZZAHL1	114
GANZZAHL2	114
GANZR	114