Programmieren mit TurboPascal

Lehrhefte für die Schule

Diplominformatiker Volkmar Heinig www.computerbildung.de

Lehrhefte für die Schule

Alle wichtigen Funktionen von WORD (46 Seiten A4, Abbildungen, Tabellen, Übersichten)

Eine Übersicht für InternetAnfänger (ca. 60 Seiten A4, Abbildungen, Tabellen, Hilfetexte)

Alle wichtigen Funktionen von EXCEL (47 Seiten A4, Abbildungen, Tabellen, Beispiele)

Arbeitsmaterialien für die Schule, Anleitungen mit Word, Excel, Powerpoint und Internet (125 Seiten, A4)

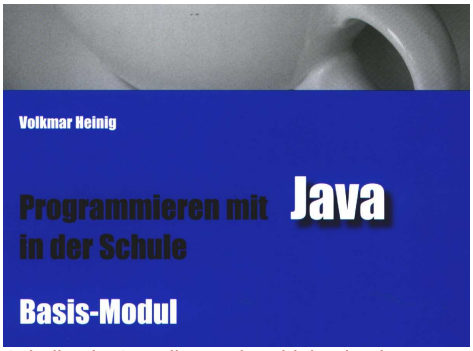
Turbo Pascal Sprachbeschreibung - gegliederte und systematische Sprachübersicht (31 Seiten, A4)

Systemaufbau des Computers und WINDOWS (37 Seiten A4, Abbildungen, Tabellen, Hilfetexte, Übersichten

Was ist eine Datenbank Einstieg in ACCESS (32 Seiten A4, Abbildungen, Beispiellösungen, Übersichten)

E-Mail-, Kalender, Adressbuch- Notizbuch, Aufgabenplanung und mehr - Bedienung des universellen Managementprogramms (A4, 115 Seiten)

Grundlagen der Grafikarbeit unter Windows- (37 Seiten A4, Abbildungen, Beispiele, Übersichten)



Schulbuch: Grundlagen der objektorientierten Programmiersprache Java, speziell für den Informatikunterricht an Schulen.
Verlag Holland + Josenhans, Stuttgart
17,80 EUR ISBN 3-7782-6022-7

www.computerbildung.de

Inhalte

1 WAS KANN PASCAL? 5

EINFÜHRUNG 5
METASPRACHE 5

2 PRINZIPIELLER PROGRAMMAUFBAU 5

3 SATZZEICHEN 6

GREP-ÄHNLICHE JOKERZEICHEN 6

4 SCHLÜSSELWÖRTER (DIREKTIVEN) 7

5 OPERATOREN 8

RANGFOLGE DER OPERATOREN 8

6 VARIABLEN & DATENTYPEN 9

VEREINBARUNGSMÖGLICHKEITEN 9
BASIS-DATENTYPEN 9
VORDEFINIERTE VARIABLEN- UND TYPNAMEN 9

7 PROGRAMMKONSTRUKTE 11

8 EIN-/AUSGABEARBEIT 11

9 DATEIARBEIT 12

10 UMWANDLUNGSROUTINEN 13

11 MATHEMATISCHE FUNKTIONEN 13

12 ZEICHENKETTENARBEIT 14

13 SPEICHEROPERATIONEN 14

14 BILSCHIRMATTRIBUTE UND TEXTFENSTERGESTALTUNG 15

15 ARBEIT IM GRAFIKMODUS 16

16 SYSTEMUMGEBUNG UND HARDWARENAHE ROUTINEN (BIOS/DOS) 19

17 PROZEßMANAGEMENT 19

18 ZEITROUTINEN 20

19 WEITERE SPRACHELEMENTE 20

20 OVERLAY-ARBEIT 21

21 UNITS 21

UNIT-PRINZIP 21
UNIT- NAMEN 22

22 FEHLERAUSWERTUNG 22

DOSERROR 22
FEHLER-CODES UND -MELDUNGEN IM GRAFIKMODUS 22
SYSTEMFEHLER FUNKTIONSCODES 23
SYSTEMFEHLER RÜCKGABEWERTE 23

23 PREPROZESSORANWEISUNGEN 24

ALLGEMEINE KONSTANTEN 26

TEXTGESTALTUNG IM GRAFIKMODUS 26

KB... TASTATURSTATUS-MASKEN 26

FUNKTIONSTASTENCODES 26

SHIFT-FUNKTIONSTASTENCODES 27

ALT-FUNKTIONSTASTENCODES 27

ALT-BUCHSTABEN TASTENCODES 27

ALT-ZAHL TASTENCODES 27

CTRL-FUNKTIONSTASTENCODES 27

SPEZIELLE TASTENCODES 28

BITBLT-OPERATOREN (PUTIMAGE) 28

KONSTANTEN FÜR DIE DATEIATTRIBUTE (GETFATTR) 28

GRAFIKTREIBER 28

GRAFIKMODI DER GRAFIKTREIBER 29

FÜLLMUSTER (SETFILLSTYLE) 29

FARBKONSTANTEN(SETPALETTE) 30

KONSTANTEN ALS PARAMETER FÜR TEXTMODE 30

LINIENARTEN UND -BREITEN (SETLINESTYLE) 30

KONSTANTEN FÜR BAR3D-AUFRUFE 30

KONSTANTEN FÜR CLIPPING 30

KONSTANTEN FÜR DIE DATEIMODI 31

KONSTANTEN FÜR DAS FLAG-REGISTER (INTR) 31

KONSTANTEN FÜR DAS FLAG-REGISTER (SETTEXTJUSTIFY) 31

KONSTANTEN DER UNIT OVERLAY (OVRREADFUNC) 31

TYPISIERTE KONSTANTEN FÜR DEN OVERLAY-MANAGER 31

STRING-KONSTANTEN 32

TYPISIERTE KONSTANTEN 32

1 Was kann PASCAL?

Einführung

TurboPascal ist der am weitesten verbreitete Pascal-Compiler für Personalcomputer nach dem MS-DOS-Standard. Diese integrierte Entwicklungsumgebung revolutionierte bereits nach ihrem Erscheinen die Technik der Programmentwicklung indem Editor, Compiler und Linker unter einer Oberfläche zusammengefasst wurden. Die enthaltenen Prozeduren und Funktionen erleichtern die Programmentwicklung und bieten schon fast fertige Problemlösungen an. Wenn Sie auch Programme für MS-Windows entwickeln möchten, finden Sie in Borland Pascal Version 7 eine geeignete Entwicklungsumgebung.
Die eigentliche Sprache Pascal will aber erlernt sein. Diese Schrift soll dafür eine Hilfestellung bieten. Die Anweisungen, Deklarationen, Konstanten, Typen, Prozeduren Funktionen und Unterprogramme. wurden nach Gruppen geordnet in einer Übersicht zusammengestellt. Die Bedienung des TurboPascal-Bedienoberfläche steht nicht im Vordergrund. Die Schrift ist als Nachschlagewerk gedacht und soll Programmiereinsteigern oder -umsteigern helfen die benötigten Anweisungen schnell herauszufinden. Grundlage der Erläuterungen bilden die integrierten Hilfen aus TurboPascal. Die Verwendung der Sprachübersicht wird Sie befähigen, schneller als bisher viele praktische Programme zu erstellen.

Metasprache

In dieser Schrift wird folgende Schreibweise verwendet:

GROSS *Standardbezeichner von Turbo-Pascal*
klein *private Bezeichner (die Sie selbst festlegen)*

2 Prinzipieller Programmaufbau

Die generelle Struktur eines Programms ist:

PROGRAM programm-name; *Programmkopf*
deklarationen; *Deklarationen mit Hilfe von Schlüsselwörtern*
BEGIN
anweisungen; *Einzelanweisungen und Unterdeklarationen*
END. *Programmfuß*

3 Satzzeichen

;	*Schließt Anweisungszeilen und Anweisungsblöcke ab.*
:	*Schließt den Label-Name ab.*
.	*Beendet den für den Compiler zu interpretierenden Quellcode.*
..	*Bedeutet: Wertebereich bei der ARRAY-Definition*
'	*Markiert Anfang und Ende eines Strings.*
"	*Markiert Anfang und Ende eines Strings.*
#	*Interpretiert den folgenden Wert als Dezimalwert (ASCII-Tabelle).*
$	*Interpretiert den folgenden numerischen Wert hexadezimal.*
h	*Interpretiert den vorangestellten numerischen Wert hexadezimal.*
b	*Interpretiert den vorangestellten numerischen Wert binär.*
o	*Interpretiert den vorangestellten numerischen Wert oktal.*
(*	*Öffnendes Kommentarzeichen*
{	*Öffnendes Kommentarzeichen*
***)**	*Schließendes Kommentarzeichen*
}	*Schließendes Kommentarzeichen*
{$befehl}	*Preprozessoranweisung - Compiler-Befehl, weist den Compiler während der Quelltext-Übersetzung an, bestimmte Operationen auszuführen.*

GREP-ähnliche Jokerzeichen

^	*Ein Zirkumflex am Anfang eines Strings gibt den Anfang einer Zeile an.*
$	*Ein Dollarzeichen am Ende eines Ausdrucks gibt das Zeilenende an.*
.	*Ein Punkt steht für ein beliebiges Zeichen.*
*****	*Ein Stern nach einem Zeichen steht für eine beliebige Anzahl von Vorkommen (einschl. Null) dieses Zeichens, z.B. steht bo* für bot, b, boo und be*
+	*Ein + nach einem Zeichen steht für eine beliebige Anzahl von Vorkommen größer Null dieses Zeichens. Z.B steht bo+ für bot und boo, aber nicht für be oder b.*
[ ]	*Zeichen in eckigen Klammern stehen für jedes Zeichen innerhalb der Klammern, aber für kein anderes (z.B. [bot] steht für b, o, oder t).*
[^]	*Ein Zirkumflex am Anfang eines Strings, der in eckigen Klammern steht, bedeutet logisch »nicht«. Z.B. [^bot] steht für jedes Zeichen, ausgenommen b, o oder t.*
[-]	*Ein Bindestrich innerhalb von eckigen Klammern kennzeichnet einen Zeichenbereich, z.B. [b-o] steht für jedes Zeichen von b bis o.*
****	*Ein Backslash vor einem Jokerzeichen weist TurboPascal an, dieses Zeichen buchstäblich und nicht als Jokerzeichen zu nehmen. Z.B. steht \^ für ^ und nicht für den Zeilenanfang.*

4 Schlüsselwörter (Direktiven)

VAR	*Leitet Variablen-Deklarationen ein, wobei Bezeichnern Datentypen und Speicherbereiche zugewiesen werden.*
TYPE	*Vereinbart Bezeichner für private Datentypen.*
CONST	*Vereinbart einen Bezeichner, der innerhalb des Programms für einen konstanten Wert steht.*
BEGIN	*Die Verbundanweisung leitet einen Anweisungsblock ein.*
END	*Beendet eines der drei folgenden Konstrukte: BEGIN,CASE,RECORD.*
PROGRAM	*Leitet das Programm mit Name und Parametern ein.(Programmkopf)*
PROCEDURE	*Leitet die Deklaration eines Unterprogramms ein.*
IMPLEMENTATION	*Leitet den Teil einer Unit ein, in welchem sich die öffentlichen Funktions- und Prozedurblöcke befinden.*
INTERFACE	*Legt fest, welche Bestandteile einer Unit öffentlich, d.h. für andere Module zugänglich sind.*
label	*Markiert Anweisungen mit entsprechendem Label.*
USES	*Leitet die Liste der Programmmodule (Units) ein, welche vom aktuellen Modul importiert werden sollen.*
FUNCTION	*Deklaration einer Funktion, welche einen Wert zurückgibt.*
UNIT	*Leitet ein separates Programmmodul ein.*
GOTO	*Führt einen Sprung zum angegebenen Label im aktuellen Block aus.*
EXTERNAL	*External-Deklarationen ermöglichen die Einbindung getrennt kompilierter Prozeduren und Funktionen (als .OBJ-Dateien), die in Maschinensprache geschrieben sind.*
FAR	*Legt das Aufrufmodell von Routinen derart fest, dass diese von anderen Code-Segmenten angesprungen werden kann.*
NEAR	*Routinen, die als "near" codiert sind, können nur von innerhalb des gleichen Codesegments angesprungen werden.*
FORWARD	*Realisiert die Definition von Programmteilen, die sich gegenseitig aufrufen (Flip-Flop-Modell).*
INTERRUPT	*Definiert eine Prozedur derart, dass sie als eine, über einen Interrupt aufrufbare, klassifiziert wird.*
bezeichner	*Private oder öffentliche (vordefinierten) Namen dienen dem eindeutigen Zuordnen folgender Direktiven: Programme, Units, Labels, Konstanten, Typen, Variablen, Prozeduren, Funktionen, Felder von Records.*
INLINE	*Bei der Deklaration einer Prozedur oder Funktion verwendet, so werden die folgenden Routinen wie Assembler-Makros behandelt.*
VIRTUAL	*Vereinbart für eine Routine eine virtuelle Laufzeitbindung.*

5 Operatoren

& *Reservierte Wörter übergehen und wie selbstdefinierte Bezeichner behandeln.*
( ) *Teilausdrücke werden zuerst ausgewertet.*
***** *Multiplikation*
+ *Addition, Strings verbinden, Zahl in positive Zahl umwandeln*
- *Subtraktion, Zahl in negative Zahl umwandeln, Ausdruck negieren*
. *Strukturauswahl xxx.yyy, Zugriff auf Felder eines Records*
/ *Division*
DIV *Ganzzahlige Division*
MOD *Modulo-Division (Rest einer ganzzahligen Division)*
:= *Zuweisungs-Operator, ersetzt den momentanen Wert einer Variablen durch einen neuen Wert.*
= *gleich (auch für TYPE)*
<> *ungleich*
< *kleiner als*
> *größer als*
<= *kleiner oder gleich*
>= *größer oder gleich*
[ ] *Speicher-Referenz*
NOT *Bitweise Negation, Logische Negation*
AND *Bitweises AND, Logisches AND*
OR *Bitweises ODER, Logisches ODER*
XOR *Bitweise Antivalenz, Logische Antivalenz*
SHL *Bitweises Linksschieben*
SHR *Bitweises Rechtsschieben*
IN *Element von*
LOW *Liefert die acht niederwertigen Bits des nachfolgenden konstanten Ausdruckes (Word)*
HIGH *Liefert die acht höherwertigen Bits des nachfolgenden konstanten Ausdruckes (Word)*
@ *Zuordnen einer Zeigervariablen*

Rangfolge der Operatoren

Rangstufe	*hoch .....niedrig*
am höchsten	**@ NOT**
	*** / DIV MOD AND SHL SHR**
	+ - OR XOR
am niedrigsten	**= <> < > <= >= IN**

Teilausdrücke können in runde Klammern eingeschlossen werden, um die Abarbeitungsreihenfolge zu ändern.

Diplominformatiker Volkmar Heinig ◆ Berlin ◆ www.computerbildung.de

6 Variablen & Datentypen

Vereinbarungsmöglichkeiten

ARRAY *Feld gleichartiger Komponenten*
ABSOLUTE *Deklaration eines Variablentyps mit fixer Speicheradresse*
RECORD *Beinhaltet eine Anzahl von Komponenten ("Feldern"), auch verschiedenen Typs.*
SET *Vereinbart die Elemente einer Menge des durch Elemententyp angegebenen Typs.*

Basis-Datentypen

REAL	*2.9e-39..1.7e38*	*6 Byte*	*11-12 prec*
SINGLE	*1.5e-45..3.4e38*	*4 Byte*	*7-8 prec*
DOUBLE	*5.0e-324..1.7e308*	*8 Byte*	*15-16 prec*
EXTENDED	*3.4e-4932..1.1e4932*	*10 Byte*	*19-20 prec*
COMP	*-9.2e18..9.2e18*	*8 Byte*	*19-20 prec*
SHORTINT	*-128..127*	*8 Bit*	
INTEGER	*-32768..32767*	*16 Bit*	
LONGINT	*-2147483648..2147483647*	*32 Bit*	
BYTE	*0..255*	*8 Bit*	
WORD	*0..65535*	*16 Bit*	
STRING	*0-255 (nach ASCII)*	*Zeichenfolge dynamischer Länge, max. 256 Zeichen*	
CHAR	*0-255 (nach ASCII)*	*1 Zeichen*	
BOOLEAN	*0-1*	*Bit (TRUE oder FALSE)*	

Vordefinierte Variablen- und Typnamen

POINTER	*Untypisierten Zeiger, der mit jedem anderen Zeigertyp kompatibel ist.*
REGISTERS	*Record-Typ zur Übergabe der Prozessor-Register*
SAVECTRLBREAK	*Status der CTRL-BREAK-Überprüfung durch DOS*
ARCCOORDSTYPE	*vordefinierter Type der Unit GRAPH.TPU*
FILLPATTERNTYPE	*- " " -*
FILLSETTINGSTYPE	*- " " -*
LINESETTINGSTYPE	*- " " -*
PALETTETYPE	*- " " -*
POINTTYPE	*- " " -*
TEXTATTR	*Speichert die aktuellen Textattribute.*
WINEURIN	*Speichert die absoluten Koordinaten der oberen linken Ecke des momentan aktiven Textfensters.*
WINEURAX	*Speichert die absoluten Koordinaten der unteren rechten Ecke des momentan aktiven Textfensters.*
TEXTSETTINGSTYPE	
VIEWPORTTYPE	*Record-Typ zum Speichern der Textfensterkoordinaten.*
MAXCOLORS	
DATETIME	
SEARCHREC	*Record-Typ für das Durchsuchen eines Datei-Verzeichnisses*
COMSTR	*string [127]*
PATHSTR	*string [79] Record-Typ für Pfadname*
DIRSTR	*string [67] Record-Typ für Verzeichnisname*
NAMESTR	*string [8] Record-Typ für Dateiname*
EXTSTR	*string [4] Dateityp*
DIRECTVIDEO	*Variable, legt fest, ob Write und WriteLn direkt in den Bildschirmspeicher schreiben oder nicht.*
LONGREC	*Erlaubt, auf das niederwertige (Lo) und höherwertige (Hi) Word eines LongInt zuzugreifen.*
WORDREC	*Erlaubt den Zugriff auf das nieder(Lo)- und höherwertige(Hi) Byte eines Word.*
MAXBUFMEM	*Gibt den maximalen Speicherbereich an, der Cache-Puffern zugewiesen werden kann.*
PTRREC	*Recordtyp, ermöglicht den Zugriff auf Offset- und Segment- wert (SEG) eines Zeigers.*
GRAPHGETMEMPTR	*Zeiger in den Grafikheap*
GRAPHFREEMEMPTR	*Zeiger in den Grafikheap*
OVRRE-SULT:INTEGER	*vordefinierte Variable der Unit OVR.TPU*
OVRFILEMODE: BYTE	*-"-*
OVRLOAD-COUNT:WORD	*-"-*

OVRREADBUF:	*-"-*
OVRREADFUNC	*-"-*
OVRTRAP-COUNT:WORD	*-"-*
OVRREADFUNC	*vordefinierter Type der Unit OVR.TPU*
INPUT:TEXT	*Standard-Eingabedatei*
OUTPUT:TEXT	*Standard-Ausgabedatei*
TEXTREC	*Recordtyp als internes Format für Variablen vom Typ TEXT*
TEXTBUF	*Recordtyp wie folgt: ARRAY[0..127] OF CHAR*
LST	*Textdatei an LPT1:*
FILE	*Ein Dateityp FILE besteht aus einer linearen Folge anderer Typkomponenten*
FILEREC	*Recordtyp für typisierte und untypisierte Dateien*

7 Programmkonstrukte

IF	*Dieses Konstrukt leitet den Vergleichsteil des bedingten Sprungs ein.*
THEN	*Leitet den Ausführungsteil der IF-Anweisung nach "wahrem" Vergleich ein.*
ELSE	*Der alternative Ausführungsweg eines mit IF oder CASE aufgebauten Konstrukts wird eingeleitet.*
CASE	*Ein Ausdruck wird hinsichtlich seines Wahrheitswertes mit einer beliebig langen Liste von Konstanten verglichen. Bei Erfüllung wird zur jeweiligen Anweisung verzweigt.*
OF	*Dieses Wort wird bei der Deklaration von Arrays, Mengen und Dateivariablen sowie für CASE-Konstrukte verwendet.*
WHILE	*Die folgenden Anweisungen werden ausgeführt (bzw. wiederholt), wenn der am Anfang geprüfte Ausdruck den Wert TRUE (wahr) besitzt.*
REPEAT	*Die folgenden Anweisungen werden 1mal ausgeführt. Wenn der am Ende geprüfte Ausdruck den Wert FALSE (falsch) besitzt, wird der Anweisungsblock wiederholt.*
UNTIL	*Schließt die zu wiederholenden Anweisungen des REPEAT-Blockes ein.*
FOR	*Die folgenden Anweisungen werden wiederholt, bis eine Laufvariable einen Zielwert erreicht hat.*
TO	*Die Laufvariable der FOR-Schleife wird bei jedem Schleifendurchlauf um eins hochgezählt.*
DOWNTO	*Der Zielwert der FOR-Schleife ist kleiner als der Start-Wert (Herabzählen).*
DO	*Leitet den eigentlichen Befehlsteil bei Schleifen ein.*

8 Ein-/Ausgabearbeit

READKEY	*Liest ein Character aus dem Tastaturpuffer, ohne es auf dem Bildschirm auszugeben.*
KEYPRESSED	*Prüft die Tastatur und liefert TRUE zurück, wenn der Tastaturpuffer noch Zeichen enthält.*

READ *Liest eine oder mehrere Komponenten aus einer Datei, im Standardfall von der Tastatur.*
READLN *Führt einen Aufruf von READ aus und springt dann zum Anfang der nächsten Zeile.*
WRITE *Schreibt eine oder mehrere Komponenten in eine Datei, im Standardfall auf den Bildschirm.*
WRITELN *Ausgabe wie WRITE, hängt aber ein Zeilenendezeichen an.*
PRINTSTR *Gibt über die DOS-Funktion 40H den String s auf dem Standard-Ausgabegerät aus. (Schneller als WRITE)*
IORESULT *Liefert den Status der letzten Ein-/Ausgabe-Operation als Integer zurück*

9 Dateiarbeit

ASSIGN *Ordnet einer Datei-Variablen eine externe Datei zu.*
ASSIGNCRT *Ordnet dem Bildschirm (CRT) eine Textdatei zu.*
BLOCKREAD *Liest einen oder mehrere Records einer Datei und schreibt in eine Puffervariable (Speicherbereich)*
BLOCKWRITE *Schreibt einen oder mehrere Records aus einer Puffervariable in eine Datei.*
CHDIR *Wechselt das aktuelle Verzeichnis.*
CLOSE *Schließt die durch f angegebene Datei.*
EOF *Prüft, ob das Ende einer Datei erreicht ist.*
EOLN *Prüft, ob das Zeilenende in einer Textdatei erreicht ist.*
ERASE *Löscht eine externe Datei beliebigen Typs.*
FEXPAND *Erweitert einen unvollständig angegebenen Dateinamen um den Suchweg zum momentan gesetzten Verzeichnis.*
FILEPOS *Liefert die aktuelle Lese-/Schreibposition innerhalb einer Datei.*
FILESIZE *Liefert die Größe einer Datei in Bytes zurück.*
FINDFIRST *Durchsucht das momentane bzw. durch Path angegebene Verzeichnis nach dem ersten Vorkommen eines Dateieintrags.*
FINDNEXT *Setzt eine mit FINDFIRST begonnene Suche fort.*
FSEARCH *Sucht eine Datei in einer Pfadliste.*
FSPLIT *Zerlegt einen vollständigen Dateinamen in seine drei Komponenten "Verzeichnispfad", "Name" und "Suffix".*
GETDIR *Ermittelt das momentan gesetzte Verzeichnis eines Laufwerks.*
GETFATTR *Liefert die Attribute einer Datei zurück.*
SETFATTR *Setzt die Attribute einer Datei.*
GETFTIME *Liefert Datum und Uhrzeit der letzten Veränderung einer Datei.*
SETFTIME *Setzt Datum und Uhrzeit der letzten Veränderung einer Datei.*
MKDIR *Erzeugt ein Unterverzeichnis.*
RENAME *Gibt einer externen Datei einen anderen Name.*
APPEND *Öffnet eine existierende Datei ab Dateiende (Anhängen von Daten).*
RESET *Öffnet eine existierende Datei ab Dateibeginn.*
REWRITE *Eröffnet eine neue Datei.*
TRUNCATE *Schneidet eine Datei an der aktuellen Position ab (EOF).*
RMDIR *Löscht ein leeres Unterverzeichnis.*
SEEK *Setzt den Positionszeiger in der durch f bezeichneten Datei.*
SEEKEOF *Prüft, ob sich zwischen der momentanen Position des Dateizeigers und dem Dateiende noch "lesbare" Daten befinden.*

SEEKEOLN *Prüft, ob sich zwischen der momentanen Position des Dateizeigers und dem Zeilenendezeichen noch "lesbare" Daten befinden.*
SETTEXTBUF *Weist einer Textdatei-Variablen einen Puffer zu.*
SETVERIFY *Setzt das Verify-Flag von DOS und legt so fest, ob DOS die geschriebenen Disk-Sektoren noch einmal überprüft.*

10 Umwandlungsroutinen

CHR *Wandelt einen Dezimalwert nach der ASCII-Tabelle in ein Zeichen um.*
ORD *Liefert die Ordinalzahl (Aufzählungswert) des Arguments dezimal.*
STR *Konvertiert den als x übergebenen Wert in eine Zeichenfolge.*
VAL *Interpretiert die Zeichenfolge des Strings s als numerischen Wert des Typs INTEGER oder REAL.*
HI *Liefert das höherwertige Teil eines Arguments zurück.*
LO *Liefert das niederwertige Teil eines Arguments zurück.*
SWAP *Vertauscht den niederwertigen und den höherwertigen Teil eines Arguments.*
UPCASE *Konvertiert Klein- in Großbuchstaben. Die deutschen Umlaute werden nicht berücksichtigt.*

11 Mathematische Funktionen

ARCTAN *Liefert den Arcustangens des Arguments zurück.*
COS *Liefert den Cosinus des Arguments zurück.*
EXP *Liefert "e hoch" dem Argument zurück.*
LN *Liefert den natürlichen Logarithmus des Arguments.*
ABS *Die Funktion liefert den Betrag des Arguments zurück.*
FRAC *Liefert den nicht-ganzzahligen Teil des Arguments zurück.*
INT *Liefert den ganzzahligen Anteil des Arguments als REAL-Zahl zurück.*
ROUND *Rundet das (Real-)Argument auf einen ganzzahligen Wert.*
TRUNC *Wandelt einen Realwert durch Abschneiden der Nachkommastellen in einen Integerwert um.*
LONGDIV *Gibt in einem Integer-Wert das Ergebnis der Division eines LongInt-Wertes durch einen Integer-Wert zurück.*
LONGMUL *Gibt in einem LongInt das Ergebnis der Multiplikation zweier Integer zurück.*
ODD *Prüft, ob das Argument eine gerade oder ungerade Zahl darstellt.*
PI *Liefert den Wert der mathematischen Konstanten Ò als Real-Zahl zurück.*
RANDOM *Liefert eine Zufallszahl.*
SIN *Liefert den Sinus des Arguments von 0..2Ò.*
SQR *Liefert die Quadratzahl des Arguments zurück.*
SQRT *Liefert die Quadratwurzel des Arguments zurück.*

12 Zeichenkettenarbeit

CONCAT	*Verbindet zwei oder mehrere Stringteile miteinander, entspricht dem Operator +.*
COPY	*Liefert einen Teil eines Strings zurück.*
DELETE	*Löscht eine Anzahl Zeichen ab einer bestimmten Position in einem String.*
FORMATSTR	*String-Formatierungsroutine - arbeitet ähnlich wie die in C definierte Funktion vsprintf.*
INSERT	*Fügt einen String in einen anderen String ab der Position "index" ein.*
LENGTH	*Liefert die Länge des als "s" übergebenen Stringausdrucks zurück.*
NEWSTR	*Dynamische String-Routine - Kopiert eine String in einen Speicherbereich und gibt einen Zeiger darauf zurück.*
DISPOSESTR	*Gibt den Speicherplatz des von NEWSTR referenzierten Strings frei.*
POS	*Durchsucht einen String-Ausdruck nach der Position des ersten Vorkommens eines 'substr'ings.*

13 Speicheroperationen

ADDR	*Die Funktion liefert die Adresse des angegebenen Objekts.*
CSEG	*Liefert die Adresse des aktuellen Code-Segments (CS-Register) zurück.*
DISPOSE	*Gibt den einer Zeigervariablen zugeordneten Speicherplatz auf dem Heap wieder frei.*
FILLCHAR	*Füllt n aufeinanderfolgende Speicherzellen mit dem Wert eines Zeichens.*
FREEBUFMEM	*Gibt den Cache-Puffer, auf den ein Pointer zeigt, frei.*
FREEMEM	*Gibt einen Speicherbereich bestimmter Größe auf dem Heap wieder frei.*
GETBUFMEM	*Reserviert einen Cache-Puffer und gibt einen Zeiger darauf zurück.*
GETMEM	*Belegt einen Bereich auf dem Heap und weist die Startadresse einer Zeigervariablen zu - erzeugt also eine dynamische Variable.*
LOWMEMSIZE	*Gibt in 16 Byte-Blöcken die Größe des Speicher-Sicherheitsbereichs für ein Programm an.*
LOWMEMORY	*Gibt TRUE zurück, wenn eine Speicheranforderung nur unter Verwendung des Sicherheitsbereichs erfüllt werden kann.*
MARK	*Hält den obersten Eintrag des Heap in der als "p" übergebenen Variablen fest.*
MAXAVAIL	*Liefert die Umfang des größten freien Blocks auf dem Heap in*

	Bytes zurück.
MEMALLOC	*Reserviert Speicher auf dem Heap und gibt einen Zeiger auf diesen Bereich zurück.*
MEMALLOCSEG	*Reserviert einen an einer Segment-Grenze beginnenden Speicher- bereich.*
MEMAVAIL	*Liefert den Gesamtumfang des freien Platzes auf dem Heap in Bytes.*
MOVE	*Kopiert "count" Bytes von dem durch "source" angegebenen Speicher- Bereich in den durch "dest" bezeichneten Bereich.*
NEW	*Belegt einen Bereich auf dem Heap und setzt die als p übergebene Zeigervariable auf die Startadresse dieses Bereichs, erzeugt also eine dynamische Variable.*
PTR	*Konvertiert zwei Angaben für Segment und Offset in einen Wert des Typs Pointer.*
PTRREC	*Ermöglicht den Zugriff auf Offset- und Segmentwert (Seg) eines Zeigers.*
RELEASE	*Setzt den Heap auf einen zuvor mit Mark festgehaltenen Zustand zurück.*
SEG	*Liefert die Segmentadresse des angegebenen Objekts zurück.*
SETMEMTOP	*Setzt die obere Grenze des Heap-Speichers auf den durch den Zeiger MEMTOP angegebenen Wert.*
SIZEOF	*Liefert die Anzahl von Bytes zurück, die ein Argument an Speicherplatz belegt.*
SPTR	*Liefert den momentanen Wert des Stackzeigers (SP-Register) zurück.*
SSEG	*Liefert die Adresse des Stack-Segments zurück, d.h. den Wert des SS-Registers.*

14 Bilschirmattribute und Textfenstergestaltung

CLREOL	*Löscht alle Zeichen von der aktuellen Position des Cursors bis zum Zeilenende.*
CLRSCR	*Löscht den Bildschirm bzw. das Textfenster durch Überschreiben mit der momentan gesetzten Hintergrundfarbe und setzt den Cursor in die obere linke Ecke.*
DELLINE	*Löscht die Zeile des Bildschirm, in der sich der Cursor momentan befindet und rollt alle darunter liegenden Zeilen hoch.*
WINDOW	*Definiert einen Bereich des Bildschirms als Textfenster über die obere linke Ecke(X1,Y1) und die untere rechte Ecke(X2,Y2).*
GOTOXY	*Setzt die Position des Cursors innerhalb des momentanen Textfensters.*
WHEREX	*Liefert die relative Spaltenposition des Cursors zum aktuellen Textfenster.*
WHEREY	*Liefert die relative Zeilenposition des Cursors zum aktuellen Textfenster.*
HIGHVIDEO	*Setzt "hohe Intensität" für die folgende Zeichenausgaben.*
LOWVIDEO	*Setzt "niedrige Intensität" für die folgende Zeichenausgaben.*
NORMVIDEO	*Setzt das Textattribut für nachfolgende Zeichenausgaben, das beim Programmstart vereinbart war.*
TEXTBACK-	*Legt die Hintergrundfarbe für folgende Textausgaben fest.*

GROUND	
TEXTCOLOR	*Legt die Zeichenfarbe für Textausgaben fest.*
INSLINE	*Fügt eine Leerzeile an der Position des Cursors ein und rollt alle folgenden Zeilen nach unten.*
TEXTMODE	*Setzt den angegebenen Textmodus.*
LASTMODE	*(Variable) Bei jedem Aufruf von TEXTMODE wird der aktuelle Videomodus in LASTMODE gespeichert.*

15 Arbeit im Grafikmodus

ARC	*Zeichnet einen Kreisbogen bzw. Kreisausschnitt.*
BAR	*Zeichnet einen gefüllten Balken, definiert durch die linke obere Ecke (x1,y1) und die rechte untere Ecke (x2,y2).*
BAR3D	*Zeichnet einen dreidimensionalen gefüllten Balken, definiert durch die linke obere Ecke (x1,y1) und die rechte untere Ecke (x2,y2).*
CIRCLE	*Zeichnet einen Kreis.*
SETVIEWPORT	*Setzt ein Grafik-Zeichenfenster, mit der oberen linke Ecke (X1,Y1) und der unteren rechte Ecke (X2,Y2).*
CLEARDEVICE	*Löscht den gesamten Grafik-Bildschirm und setzt den Grafik-Cursor auf den Ursprung (0,0) des aktuellen Fensters. Andere Parameter des Grafikpakets werden nicht zurückgesetzt.*
CLEARVIEWPORT	*Löscht den Inhalt des aktuellen Zeichenfensters.*
CLOSEGRAPH	*Beendet den Grafikmodus. Geladene Grafiktreiber und Zeichensätze werden aus dem Hauptspeicher entfernt, danach wird der Textmodus gesetzt, der vor der Aktivierung des Grafikpakets aktiv war.*
DETECTGRAPH	*Prüft die Hardware des Computers und bestimmt, welcher Grafiktreiber beim Aufruf von INITGRAPH geladen werden soll.*
DRAWPOLY	*Zeichnet den Umriss eines Polygons.*
ELLIPSE	*Zeichnet einen gefüllten elliptischen Kreisausschnitt.*
FILLELLIPSE	*Zeichnet eine Ellipse.*
FILLPOLY	*Füllt ein Polygon mit dem momentan gesetzten Füllmuster und der aktuellen Zeichenfarbe.*
FLOODFILL	*Füllt einen umschlossenen Bereich mit dem aktuellem Füllmuster und Zeichenfarbe.*
GETARCCOORDS	*Liefert Daten über den zuletzt ausgeführten Aufruf der Prozedur ARC zurück.*
GETASPECTRATIO	*Ermittelt das physikalische Höhen-/Seitenverhältnis des Bildschirms.*
SETASPECTRATIO	*Setzt den Korrekturfaktor für das Höhen-/Seitenverhältnis des Bildschirms neu.*
GETBKCOLOR	*Liefert die momentan gesetzte Hintergrundfarbe zurück.*
SETBKCOLOR	*Setzt die Hintergrundfarbe des Bildschirms.*
GETCOLOR	*Liefert die aktuelle Zeichenfarbe zurück.*
SETCOLOR	*Legt fest, welcher Paletteneintrag als Zeichenfarbe verwendet werden soll.*
GETDEFAULTPALETTE	*Liefert die Farbpalette zurück, die bei der Initialisierung des Grafiktreibers gesetzt war.*

GETDRIVERNAME	*Liefert den Namen des momentan geladenen Grafiktreibers zurück.*
GETFILLPATTERN	*Liefert das momentan gesetzte Füllmuster zurück.*
SETFILLPATTERN	*Ermöglicht die Definition eines beliebigen Füllmusters.*
GETFILLSETTINGS	*Liefert Daten über das momentan gesetzte Füllmuster und die verwendete Farbe.*
SETFILLSTYLE	*Legt eines der 12 vordefinierten Muster und die Farbe für Aufrufe von Bar, Bar3D, FillPoly und PieSlice fest.*
SETGRAPHBUFSIZE	*Legt die Größe des Puffers für Flächenfüllungen und Polygonzüge fest.*
GETGRAPHMODE	*Ermittelt den momentan gesetzten Grafikmodus in Abhängigkeit vom verwendeten Grafiktreiber.*
SETGRAPHMODE	*Setzt den durch Mode angegebenen Grafikmodus und löscht den Bildschirm.*
GETIMAGE	*Kopiert einen rechteckigen Bildausschnitt in die als BitMap übergebene Variable.*
GETLINESETTINGS	*Liefert Daten über die momentan festgelegte Linienart.*
GETMAXCOLOR	*Liefert die höchste verwendbare Farbnummer für den momentan verwendeten Grafiktreiber und Grafikmodus.*
GETMAXMODE	*Ermittelt den maximal möglichen Wert für den Grafikmodus mit der höchsten Auflösung.*
GETMAXX	*Liefert die maximale X-Koordinate des Grafik-Bildschirms.*
GETMAXY	*Liefert die maximal Y-Koordinate des Grafik-Bildschirms.*
GETMODENAME	*Erwartet die Nummer eines Grafikmodus für den momentan aktiven Grafiktreiber und liefert den Namen dieses Modus zurück.*
GETMODERANGE	*Liefert den Wertebereich gültiger Moduswerte für den angegebenen Grafiktreiber.*
GETPALETTE	*Liefert Informationen über die momentan gesetzte Farbpalette.*
SETPALETTE	*Ändert den durch COLORNUM bezeichneten Eintrag der aktuellen Farbpalette auf die durch COLOR angegebene Farbe.*
SETRGBPALETTE	*Ändert den durch COLORNUM bezeichneten Eintrag in der aktuellen Farbpalette für den IBM-Adapter 8514 und für VGA-Karten.*
SETALLPALETTE	*Setzt alle Einträge einer Farbpalette neu.*
GETPALETTESIZE	*Liefert die Größe der aktuellen Farbpalette zurück, d.h. die Anzahl der Einträge.*
GETPIXEL	*Liefert die Farbe des Pixels mit den Koordinaten (X,Y) zurück.*
GETTEXTSETTINGS	*Liefert Informationen über Textstil, -größe, -ausrichtung und Schreibrichtung.*
GETX	*Liefert die aktuelle X-Koordinate des Grafik-Cursors zurück.*
GETY	*Liefert die aktuelle Y-Koordinate des Grafik-Cursors zurück.*
GRAPHRESULT	*Liefert den Fehler-Code der letzten Grafikoperation zurück.*
GRAPHERRORMSG	*Liefert den Text der Grafik-Fehlermeldung zurück, der dem als ErrorCode übergebenen Wert entspricht.*
GRAPHDEFAULTS	*Positioniert den Grafikcursor in die linke obere Ecke (0,0) und setzt das Grafiksystem auf Standard-Einstellungen zurück.*
IMAGESIZE	*Berechnet die Größe des Speichers, für eine Puffervariable, zum abzuspeichernden eines Bildschirmausschnittes.*
INITGRAPH	*Initialisiert einen Grafikmodus für die Bildschirmausgabe und*

	schaltet in diesen um.
SETTEXTJUSTIFY	*Bewirkt eine horizontale oder vertikale Ausrichtung von Texten.*
SETLINESTYLE	*Setzt die Linienart und -dicke für Zeichenaktionen.*
LINE	*Zeichnet eine Linie vom Punkt (x1,y1) zum Punkt (x2,y2).*
LINEREL	*Zeichnet eine Linie von der aktuellen Cursorposition zur relativen Cursorposition(x2,y2), der Grafikcursor verbleibt anschließend auf (x1,y1).*
LINETO	*Zeichnet eine Linie von der momentanen Position des Grafik-Cursors zum Punkt (x2,y2).*
MOVEREL	*Versetzt den Grafik-Cursor relativ zu seiner momentanen Position.*
MOVETO	*Bewegt den Grafik-Cursor zum Punkt (X,Y).*
SETWRITEMODE	*Legt fest, ob das Zeichnen von Linien als Verknüpfung mit dem vorherigen Bildinhalt ausgeführt wird.*
OUTTEXT	*Gibt Text an der aktuellen Position des Grafik-Cursors aus.*
OUTTEXTXY	*Gibt Text an der Position (X,Y) des Grafik-Cursors aus.*
SETTEXTJUSTIFY	*Legt die Richtung für folgende Textausgaben mit OUTTEXT und OUTTEXTXY fest.*
SETTEXTSTYLE	*Legt den Zeichensatz, die Rotation und die Größe der Textausgaben mit OUTTEXT und OUTTEXTXY fest.*
SETUSERCHARSIZE	*Setzt voneinander unabhängige Skalierungen des momentanen Zeichensatzes in X- und Y-Richtung.*
TEXTHEIGHT	*Liefert die Höhe eines Textstrings in Pixel zurück.*
TEXTWIDTH	*Liefert die Breite eines Textstrings in Pixel zurück.*
PIESLICE	*Zeichnet ein ausgefülltes Kreissegment mit dem Mittelpunkt (X,Y) und angegebenem Radius.*
PUTIMAGE	*Kopiert den Inhalt der als BitMap übergebenen Puffervariablen bitweise in einen rechteckigen Bildausschnitt.*
PUTPIXEL	*Setzt ein "Pixel" mit der festgelegten Farbe am Punkt (X,Y).*
RECTANGLE	*Zeichnet ein Rechteck mit der aktuellen Linienart und Farbe via linke, oberer Ecke und rechte, untere Ecke.*
REGISTERBGIDRIVER	*Registriert einen als .OBJ-Datei einzubindenden Grafiktreiber.*
REGISTERBGIFONT	*Registriert einen als .OBJ-Datei einzubindenden Vektor- Zeichensatz.*
RESTORECRTMODE	*Setzt den Textmodus, der vor der Aktivierung des Grafikmodus mit INITGRAPH aktiv war.*
SECTOR	*Zeichnet ein elliptisches Kreissegment.*
SETACTIVEPAGE	*Legt fest, auf welcher Grafik-Speicherseite folgende Zeichenbefehle arbeiten.*
SETVISUALPAGE	*Stellt den Inhalt eine Grafik-Speicherseite auf dem Bildschirm dar.*
INSTALLUSERDRIVER	*Erweitert das Grafikpaket um einen privaten Grafiktreiber.*
INSTALLUSERFONT	*Installiert private Vektor-Zeichensätze.*

16 Systemumgebung und hardwarenahe Routinen (BIOS/DOS)

CHECKBREAK	*Variable - Erlaubt oder verhindert den Abbruch eines Programms mittels Ctrl-Break (<Strg> + <Pause>).*
BUTTONCOUNT	*Variable - Speichert die Zahl der Knöpfe der Maus oder ist 0, wenn keine Maus installiert ist.*
CHECKEOF	*Variable - Gestattet oder verhindert die Erzeugung von Dateiende-Zeichen.*
CPU86	*Das Symbol wird definiert, um anzuzeigen, dass der Prozessor zu der 80x86 Prozessorfamilie gehört.*
CPU87	*Dieses Symbol wird nur definiert, wenn auch ein entsprechender Coprozessor zur Übersetzungszeit vorhanden ist.*
DELAY	*Die weitere Programmausführung wartet die angegebene Anzahl von Millisekunden.*
DISKFREE	*Liefert die Anzahl der freien Bytes auf dem angegebenen Laufwerk zurück.*
DISKSIZE	*Liefert die Gesamtgröße des Datenträgers im angegebenen Laufwerk zurück.*
DOSVERSION	*Liefert die Versionsnummer von DOS zurück.*
DOUBLEDELAY	*Variable - Zeitabstand zwischen zwei Maus-Klicks (Doppelklick)*
ENVCOUNT	*Liefert die Anzahl der Einträge in der DOS Environment-Tabelle zurück.*
ENVSTR	*Liefert einen Eintrag der DOS Environment-Tabelle zurück.*
GETENV	*Sucht die DOS-Tabelle Environment nach dem angegebenen Eintrag ab.*
GETALTCHAR	*Ermittelt, welche [Alt]-Tastenkombination den Wert von Key-Code liefert, und gibt das Zeichen zurück, das auf dieser Taste liegt.*
GETALTCODE	*Gibt den Wert zurück, den auch die Tastatur liefert, wenn eine Kombination aus [Alt] und einem Zeichen gedrückt würde.*
GETCBREAK	*Ermittelt, bei welchen Operationen DOS auf Ctrl-Break prüft.*
SETCBREAK	*Setzt das Break-Flag von DOS mit dem als Break übergebenen Wert.*
GETINTVEC	*Ermittelt den Zeiger auf die Interrupt-Behandlungsroutine eines angegebenen Interrupts.*
SETINTVEC	*Setzt den durch INTNO angegebenen Interrupt-Vektor des Systems auf die Adresse, auf die der als Vector übergebene Zeiger zeigt.*
LST	*Die Unit PRINTER definiert eine Textdatei namens LST und bindet sie an den Druckerport LPT1.*
INITSYSERROR	*Übernimmt Interruptvektoren und hebt den Ctrl-Break-Status in DOS auf.*
DONESYSERROR	*Beendet die Fehlerbehandlung, indem die Interruptvektoren und der Ctrl-Break-Status restauriert werden.*

17 Prozeßmanagement

DOSEXITCODE *Liefert den Exitcode eines Subprozesses, d.h. eines Programms, das mit Exec gestartet wurde.*
EXEC *Führt ein Programm via COMMAND als Subprozess aus.*
SWAPVECTORS *Vertauscht die vom SYSTEM belegten Interrupt-Vektoren mit den entsprechenden Variablen der Unit. Diese Routine sollte vor und nach dem Aufruf anderer Programme via Exec benutzt werden.*
EXIT *Verlässt den momentanen Block mit sofortiger Wirkung.*
HALT *Bricht die Ausführung des Programms ab und führt einen Rücksprung zur DOS-Kommandoebene bzw. aufrufenden Programm durch.*
INTR *Führt den durch INTNO angegebenen Software-Interrupt aus.*
KEEP *Beendet ein Programm und macht es speicherresident.*
RUNERROR *Erzeugt einen gewünschten Laufzeitfehler, der das Programm definiert abbricht.*

18 Zeitroutinen

GETDATE *Ermittelt das aktuelle Datum des Systems.*
SETDATE *Setzt das Datum des Betriebssystems.*
GETTIME *Ermittelt die momentan gesetzte Uhrzeit des Systems.*
SETTIME *Setzt die Uhrzeit des Systems.*
GETFTIME *Ermittelt Datum und Uhrzeit der letzten Veränderung einer Datei.*
PACKTIME *Konvertiert einen Record des Typs DATETIME in eine 4 Byte lange Struktur (formal: LONGINT) für SETFTIME.*
SETDATE *Setzt das Kalenderdatum des Systems.*
SETFTIME *Setzt Modifikations-Datum und -Uhrzeit einer offenen Datei.*
SETTIME *Setzt die Uhrzeit des Systems.*
UNPACKTIME *Konvertiert eine gepackte Struktur von 4 Bytes (formal:LONGINT) in einen Record des Typs DATETIME. Derartige Strukturen werden von GETFTIME, FINDFIRST, oder FINDNEXT erzeugt.*
PACKTIME *Konvertiert einen Record des Typs DATETIME in das gepackte Datums-/Uhrzeitformat von DOS.*

19 Weitere Sprachelemente

DEC *Zählt eine Variable um einen bestimmten Wert herunter.*
INC *Erhöht eine Variable um einen bestimmten Wert.*
ASM *Leitet Anweisung(en) in Assembler-Syntax ein.*
INLINE *Datenworte(Bytefolgen) werden direkt in den vom Compiler erzeugten Code eingefügt.*
SOUND *Aktiviert den eingebauten Lautsprecher mit einer angegebenen Frequenz.*
NOSOUND *Schaltet den eingebauten Lautsprecher (nach einem Aufruf von Sound) wieder ab.*
PARAMCOUNT *Liefert die Anzahl der Kommandozeilen-Parameter zurück, die dem Programm bei dessen Aufruf übergeben werden.*
PARAMSTR *Liefert den mit 'index' bezeichneten Kommandozeilen-Parameter als String zurück.*

PRED	*Liefert den Vorgänger des Arguments eines beliebigen Typs zurück.*
SUCC	*Liefert den Nachfolger des Arguments eines beliebigen Typs zurück.*
RANDOMIZE	*Initialisiert den Zufallszahlengenerator.*
WITH	*Ermöglicht, dass die Felder einer Record-Variablen im folgenden Anweisungsblock nur mit ihren Feldnamen angesprochen werden können.*

20 Overlay-Arbeit

OVRCLEARBUF	*Entfernt alle geladenen Overlay-Units aus dem Hauptspeicher und erzwingt ein Nachladen der entsprechenden Codesegmente aus der .OVR-Datei bzw. dem EMS-Speicher.*
OVRGETBUF	*Liefert die momentane Größe des Overlay-Puffers (in Bytes) zurück.*
OVGETRETRY	*Gibt die aktuelle Größe des Bewährungsbereiches im Overlay-Puffer zurück.*
OVRINIT	*Initialisiert die Overlay-Verwaltung und öffnet die .OVR-Datei des Programms.*
OVRINITEMS	*Prüft, ob das System mit EMS-Speicher ausreichender Kapazität ausgerüstet ist und lädt ggf. die gesamte .OVR-Datei in diesen Speicherbereich.*
OVRSETBUF	*Legt die Größe des Overlay-Puffers durch Verschiebung des Heap-Beginns explizit fest.*
OVRSETRETRY	*Stellt die Größe des Bewährungsbereiches im Overlay-Puffer ein.*

21 Units

Unit-Prinzip

Eine Unit ist eine Sammlung von Konstanten, Datentypen, Variablen, Prozeduren und Funktionen. Jede Unit stellt praktisch ein abgeschlossenes Programm dar und wird separat kompiliert. Die Aufnahme in ein Programm erfolgt mit einer USES-Anweisung. TurboPascal stellt acht Standard-Units zur Verfügung.

Unit- Namen

CRT.TPU *Die Routinen dieser Unit ermöglichen die direkte Kontrolle des Bildschirms, der Tastatur und des Tongenerators.*

DOS.TPU *Die Unit bildet die Schnittstelle zu den MS-DOS-Funktionen.*

GRAPH.TPU *Die Unit beinhaltet die Routinen zur Arbeit im Grafikmodus.*

OVERLAY.TPU *Die Unit enthält Prozeduren, Funktionen und Symbole zur Overlayverwaltung.*

PRINTER.TPU *Die Unit definiert eine Textdatei namens LST.*

SYSTEM.TPU *Die Unit System enthält hardwarenahe Routinen, sie wird automatisch in jedes Programm eingebunden, eine USES-Anweisung ist dafür nicht zugelassen.*

22 Fehlerauswertung

DOSERROR

Variable, die hier gespeicherten Werte sind Fehlercodes von DOS.
DOSERROR = 0 *heißt "fehlerfreie Ausführung".*

2	*File not found*	*Datei nicht gefunden*
3	*Path not found*	*Suchweg nicht gefunden*
5	*Access denied*	*Zugriff verweigert*
6	*Invalid handle*	*Handle ungültig*
8	*Not enough memory*	*Nicht genügend Speicher*
10	*Invalid environment*	*Ungültige Environment-Variablen*
11	*Invalid format*	*Ungültiges Befehlsformat*
18	*No more files*	*Keine weiteren Dateieinträge*

Fehler-Codes und -Meldungen im Grafikmodus

0	**GROK**	*No error*	*fehlerfreie Ausführung*
-1	**GRNOINITGRAPH**	*BGI graphics not installed*	*BGI-Treiber nicht geladen*
-2	**GRNOTDE-**	*Graphics hardware not de-*	*Grafikkarte nicht erkannt*

	TECTED	*tected*	
-3	**GRFILENOT-FOUND**	*Device driver file not found*	*Grafiktreiber(-datei) nicht gefunden*
-4	**GRINVALID-DRIVER**	*Invalid device driver file*	*Grafiktreiber defekt*
-5	**GRNOLOAEUREM**	*Not enough memory to load driver*	*nicht genug Speicher, um Treiber zu laden*
-6	**GRNOSCANMEM**	*Out of memory in scan fill*	*nicht genug Speicher für Zeichenoperation*
-7	**GRNOFLOOEUREM**	*Out of memory in flood fill*	*nicht genug Speicher für Fülloperation*
-8	**GRFONTNOT-FOUND**	*Font file not found*	*Zeichensatz-Datei nicht gefunden*
-9	**GRNOFONTMEM**	*Not enough memory to load font*	*nicht genug Speicher für Zeichensatz*
-10	**GRINVALIEU-RODE**	*Invalid graphics mode for selected driver*	*Grafikmodus wird vom Treiber nicht unterstützt*
-11	**GRERROR**	*Graphics error (generic error)*	*nicht näher klassifizierbarer Fehler*
-12	**GRIOERROR**	*Graphics I/O error*	*Fehler bei Daten-Ein-/Ausgabe*
-13	**GRINVALIDFONT**	*Invalid font file*	*Zeichensatz-Datei ungültig*
-14	**GRINVALID-FONTNUM**	*Invalid font number*	*Kennziffer für Zeichensatz nicht definiert*

Systemfehler Funktionscodes

0..12 *DOS kritischer Fehler*
13 *FAT im Speicher fehlerhaft*
14 *Lesefehler Gerät*
15 *Laufwerk Swap*

Systemfehler Rückgabewerte

0 *Wiederholung angefordert*
1 *Abbruch angefordert*

23 Preprozessoranweisungen

{$A} *Align Data - Legt fest, wie Compiler und Linker Variablen mit einem Platzbedarf von mehr als einem Byte im Speicher ablegen.*

{$B} *Legt fest, wie mit AND und OR verbundene boolesche Ausdrücke ausgewertet werden.*

{$D} *Debugger-Informationen - Legt fest, ob der Compiler Informationen für den Debugger erzeugt oder nicht.*

{$DEFINE} *Direktive zur bedingten Kompilierung - Definiert das durch Name angegebene Symbol.*

**{$E}** *80x87-Emulator - Legt fest, welche Routinen im Modus {$N+} zur Steuerung des Coprozessors in das Programm eingebaut werden sollen.*

{$ELSE} *Direktive zur bedingten Kompilierung für die alternative Verzweigung.*

**{$ENDIF}** *Direktive zur bedingten Kompilierung - Schließt ein mit {$IF...} begonnenes Konstrukt ab.*

{$F} *Erzwingen von FAR-Aufrufen*

{$G} *80286-Code-Erzeugung (16Bit-AT-Befehlscode)*

{$I} *Legt fest, ob Ein- und Ausgaben zur Laufzeit des Programms automatisch auf Fehler überprüft werden sollen.*

{$IFDEF} *Bewirkt die Übersetzung des folgenden Quelltextes, wenn das bezeichnete Symbol definiert ist.*

{$IFNDEF} *Bewirkt die Übersetzung des folgenden Quelltextes, wenn das bezeichnete Symbol nicht definiert ist.*

{$IFOPT} *Bestimmt die Kompilierung des nachfolgenden Quelltextes abhängig vom Stand eines Compiler-Schalters.*

**{$L}** *Legt fest, ob TurboPascal neben den über {$D} erzeugten Zeilennummern weitere Informationen über lokale Symbole in ein Modul aufnimmt oder nicht.*

{$M} *Speicherbelegung - Mit dieser Direktive können Sie angeben, wieviel Stack und Heap Ihr Programm benötigt.*

{$N} *Legt fest, ob Operationen mit Realzahlen durch Routinen der Laufzeitbibliothek oder über die direkte Ansteuerung eines numerischen Coprozessor stattfinden.*

{$O} *Legt fest ob der Compiler eine Unit overlayfähig macht oder nicht.*

{$R} *Legt fest, ob der Compiler zusätzlichen Code zur Bereichsüberprüfung generiert.*

{$S} *Legt fest, ob der Compiler zusätzlichen Prüfcode vor Belegungen des Stacks erzeugt.*

{$UNDEF} *Direktive zur bedingten Kompilierung - Löscht ein bereits bezeichnetes Symbol.*

{$V} *Legt fest, welche Prüfungen der Compiler bei der Übergabe von Strings als VAR-Parameter vornimmt.*

{$X} *Legt fest, ob die erweiterte Syntax von TurboPascal benutzt wird oder nicht.*

{$I filename} *Include-Datei (PASCAL-Quelltext) einfügen*

{$L filename} *Einbinden von Object-Dateien*

{$O unitname} *Name der Overlay Unit*

{$M stacksize, *Größen von Stack und Heap festlegen*

**heapmin,
heapmax}**

Anhang - Vordefinierte Konstanten und Typen

Allgemeine Konstanten

NIL	*Not In List - Zeiger auf NUL*
FALSE	*Wahrheitswert (Datentyp BOOLEAN)*
TRUE	*Wahrheitswert (Datentyp BOOLEAN)*
MAXINT	*größte, als INTEGER darstellbare Zahl (32 767).*
MAXLONGINT	*die größte Zahl, die sich mit dem Datentyp LONGINT darstellen lässt (2 147 483 647).*

Textgestaltung im Grafikmodus

DEFAULTFONT	**0**	*8x8 Bitmusterzeichensatz*
TRIPLEXFONT	**1**	*Vektorzeichensatz*
SMALLFONT	**2**	*Vektorzeichensatz*
SANSSERIFFONT	**3**	*Vektorzeichensatz*
GOTHICFONT	**4**	*Vektorzeichensatz*
HORIZDIR	**0**	*von links nach rechts*
VERTDIR	**1**	*von unten nach oben*
USERCHARSIZE	**0**	*benutzerdefinierte Zeichengröße*

KB... Tastaturstatus-Masken

KBRIGHTSHIFT	**$0001**	*Rechte Shift-Taste gedrückt*
KBLEFTSHIFT	**$0002**	*Linke Shift-Taste gedrückt*
KBCTRLSHIFT	**$0004**	*Ctrl- und Shift-Tasten gedrückt*
KBALTSHIFT	**$0008**	*Alt- und Shift-Tasten gedrückt*
KBSCROLLSTATE	**$0010**	*Scroll lock ein*
KBNUMSTATE	**$0020**	*Num lock ein*
KBCAPSSTATE	**$0040**	*Caps lock ein*
KBINSSTATE	**$0080**	*Insert mode ein*

Funktionstastencodes

KBF1	**$3B00**	**KBF6**	**$4000**
KBF2	**$3C00**	**KBF7**	**$4100**
KBF3	**$3D00**	**KBF8**	**$4200**
KBF4	**$3E00**	**KBF9**	**$4300**
KBF5	**$3F00**	**KBF10**	**$4400**

Shift-Funktionstastencodes

KBSHIFTF2	$5500	KBSHIFTF7	$5A00
KBSHIFTF3	$5600	KBSHIFTF8	$5B00
KBSHIFTF4	$5700	KBSHIFTF9	$5C00
KBSHIFTF5	$5800	KBSHIFTF10	$5D00

Alt-Funktionstastencodes

KBALTF1;	$6800	KBALTF6;	$6D00
KBALTF2;	$6900	KBALTF7;	$6E00
KBALTF3;	$6A00	KBALTF8;	$6F00
KBALTF4;	$6B00	KBALTF9;	$7000
KBALTF5;	$6C00	KBALTF10	$7100

Alt-Buchstaben Tastencodes

KBALTA	$1E00	KBALTN	$3100
KBALTB	$3000	KBALTO	$1800
KBALTC	$2E00	KBALTP	$1900
KBALTD	$2000	KBALTQ	$1000
KBALTE	$1200	KBALTR	$1300
KBALTF	$2100	KBALTS	$1F00
KBALTG	$2200	KBALTT	$1400
KBALTH	$2300	KBALTU	$1600
KBALTI	$1700	KBALTV	$2F00
KBALTJ	$2400	KBALTW	$1100
KBALTK	$2500	KBALTX	$2D00
KBALTL	$2600	BALTY	$1500
KBALTM	$3200	KBALTZ	$2C00

Alt-Zahl Tastencodes

KBALT1	$7800	KBALT6	$7D00
KBALT2	$7900	KBALT7	$7E00
KBALT3	$7A00	KBALT8	$7F00
KBALT4	$7B00	KBALT9	$8000
KBALT5	$7C00	KBALT0	$8100

Ctrl-Funktionstastencodes

KBCTRLF1	$5E00	KBCTRLF6	$6300
KBCTRLF2	$5F00	KBCTRLF7	$6400
KBCTRLF3	$6000	KBCTRLF8	$6500
KBCTRLF4	$6100	KBCTRLF9	$6600
KBCTRLF5	$6200	KBCTRLF10	$6700

Spezielle Tastencodes

KBALTEQUAL	**$8300**	**KBEND**	**$4F00**
KBALTMINUS	**$8200**	**KBENTER**	**$1C0D**
KBALTSPACE	**$0200**	**KBESC**	**$011B**
KBBACK	**$0E08**	**KBGRAYMINUS**	**$4A2D**
KBCTRLBACK	**$0E7F**	**KBHOME**	**$4700**
KBCTRLDEL	**$0600**	**KBINS**	**$5200**
KBCTRLEND	**$7500**	**KBLEFT**	**$4B00**
KBCTRLENTER	**$1C0A**	**KBNOKEY**	**$0000**
KBCTRLHOME	**$7700**	**KBPGDN**	**$5100**
KBCTRLINS	**$0400**	**KBPGUP**	**$4900**
KBCTRLLEFT	**$7300**	**KBGRAYPLUS**	**$4E2B**
KBCTRLPGDN	**$7600**	**KBRIGHT**	**$4D00**
KBCTRLPGUP	**$8400**	**KBSHIFTDEL**	**$0700**
KBCTRLPRTSC	**$7200**	**KBSHIFTINS**	**$0500**
KBCTRLRIGHT	**$7400**	**KBSHIFTTAB**	**$0F00**
KBDEL	**$5300**	**KBTAB**	**$0F09**
KBDOWN	**$5000**	**KBUP**	**$4800**

BITBLT-Operatoren (PUTIMAGE)

NORMALPUT	**0**	**MOV**	*ÜBERSCHREIBEN*
COPYPUT	**0**	**MOV**	
XORPUT	**1**	**XOR**	*XOR-OPERATION*
ORPUT	**2**	**OR**	*ODER-VERKNÜPFUNG*
ANDPUT	**3**	**AND**	*UND-VERKNÜPFUNG*
NOTPUT	**4**	**NOT**	*INVERTIERTES ÜBERSCHREIBEN*

Konstanten für die Dateiattribute (GETFATTR)

READONLY	**$01**
HIDDEN	**$02**
SYSFILE	**$04**
VOLUMEID	**$08**
DIRECTORY	**$10**
ARCHIVE	**$20**
ANYFILE	**$3F**

Grafiktreiber

CURRENTDRIVER	**-128** *für* **GETMODERANGE**
DETECT	**0** *automatische Erkennung*
CGA	**1**
MCGA	**2**
EGA	**3**
EGA64	**4**
EGAMONO	**5**
IBM8514	**6**
HERCMONO	**7**
ATT400	**8**
VGA	**9**
PC3270	**10**

Grafikmodi der Grafiktreiber

CGAC0	**0**	*320 x 200*	**EGALO**	**0**	*640 x 200*
CGAC1	**1**	*320 x 200*	**EGAHI**	**1**	*640 x 350*
CGAC2	**2**	*320 x 200*	**EGAMONOHI**	**3**	*640 x 350*
CGAC3	**3**	*320 x 200*	**EGA64LO**	**0**	*640 x 200*
CGAHI	**4**	*640 x 200*	**EGA64HI**	**1**	*640 x 350*
MCGAC0	**0**	*320 x 200*	**TT400C0**	**0**	*320 x 200*
MCGAC1	**1**	*320 x 200*	**ATT400C1**	**1**	*320 x 200*
MCGAC2	**2**	*320 x 200*	**ATT400C2**	**2**	*320 x 200*
MCGAC3	**3**	*320 x 200*	**ATT400C3**	**3**	*320 x 200*
MCGAMED	**4**	*640 x 200*	**ATT400MED**	**4**	*640 x 200*
MCGAHI	**5**	*640 x 480*	**ATT400HI**	**5**	*640 x 400*
IBM8514LO	**0**	*640 x 480*	**PC3270HI**	**0**	*720 x 350*
IBM8514HI	**1**	*1024 x 768*	**HERCMONOHI**	**0**	*720 x 348*
VGALO	**0**	*640 x 200*			
VGAMED	**1**	*640 x 350*			
VGAHI	**2**	*640 x 480*			

Füllmuster (SETFILLSTYLE)

EMPTYFILL	**0**	*Füllen mit Hintergrundfarbe*
SOLIDFILL	**1**	*Füllen mit der Zeichenfarbe*
LINEFILL	**2**	*---*
LTSLASHFILL	**3**	*/// *
SLASHFILL	**4**	*/// mit dicken Linien*
BKSLASHFILL	**5**	*\\\ mit dicken Linien*
LTBKSLASHFILL	**6**	**
HATCHFILL	**7**	*leicht schraffiert*
XHATCHFILL	**8**	*stark schraffiert, überkreuzend*
INTERLEAVEFILL	**9**	*abwechselnde Linien*
WIDEDOTFILL	**10**	*weit auseinanderliegende Punkte*

CLOSEDOTFILL	**11**	*eng beieinanderliegende Punkte*
USERFILL	**12**	*benutzerdefinierbares Füllmuster*

Farbkonstanten(SETPALETTE)

BLACK	**0**	**DARKGRAY**	**8**
BLUE	**1**	**LIGHTBLUE**	**9**
GREEN	**2**	**LIGHTGREEN**	**10**
CYAN	**3**	**LIGHTCYAN**	**11**
RED	**4**	**LIGHTRED**	**12**
MAGENTA	**5**	**LIGHTMAGENTA**	**13**
BROWN	**6**	**YELLOW**	**14**
LIGHTGRAY	**7**	**WHITE**	**15**

Konstanten als Parameter für TEXTMODE

BW40	**0**	*{ 40x25 S/W, CGA }*
CO40	**1**	*{ 40x25 Farbe, CGA }*
BW80	**2**	*{ 80x25 S/W, CGA }*
CO80	**3**	*{ 80x25 Farbe, CGA }*
MONO	**7**	*{ 80x25 S/W, MDA oder HGC }*
FONT8X8	**255**	*{ 43-/50-Zeilenmodus EGA/VGA }*

Linienarten und -breiten (SetLineStyle)

SOLIDLN	**0**
NORMWIDTH	**1**
DOTTEDLN	**1**
THICKWIDTH	**3**
CENTERLN	**2**
DASHEDLN	**3**
USERBITLN	**4** *(selbstdefinierbare Linienart)*

Konstanten für BAR3D-Aufrufe

TOPON	*true*	*Säule mit Deckfläche*
TOPOFF	*false*	

Konstanten für Clipping

CLIPON;	*true*

CLIPOFF *false*

Konstanten für die Dateimodi

FMCLOSED	**$D7B0**
FMINPUT	**$D7B1**
FMOUTPUT	**$D7B2**
FMINOUT	**$D7B3**

Konstanten für das Flag-Register (INTR)

FCARRY	**$0001**
FPARITY	**$0004**
FAUXILIARY	**$0010**
FZERO	**$0040**
FSIGN	**$0080**
FOVERFLOW	**$0800**

Konstanten für das Flag-Register (SETTEXTJUSTIFY)

Horizontal		*Vertikal*	
LEFTTEXT	**0**	**BOTTOMTEXT**	**0**
CENTERTEXT	**1**	**CENTERTEXT**	**1**
RIGHTTEXT	**2**	**TOPTEXT**	**2**

Konstanten der Unit Overlay (OVRREADFUNC)

OVROK	**0**	*{ kein Fehler}*
OVRERROR	**-1**	*{ allg. Fehler }*
OVRNOTFOUND	**-2**	*{ .OVR-Datei nicht gefunden }*
OVRNOMEMORY	**-3**	*{ kein Platz für den Overlay-Puffer }*
OVRIOERROR	**-4**	*{ Fehler beim Lesen der .OVR-Datei }*
OVRNOEMSDRIVER	**-5**	*{ OVRINITEMS: kein EMS-Treiber }*
OVRNOEMSMEMORY	**-6**	*{ OVRINITEMS: nicht genug EMS-Speicher }*

Typisierte Konstanten für den Overlay-Manager

OVRCODELIST: *WORD 0 { Codesegment Liste }*
OVRDEBUGPTR: *POINTER NIL { Debugger}*

OVRDOSHANDLE:	*WORD*	*0*	*{ OVR-Handle }*
OVREMSHANDLE	*WORD*	*0*	*{ OVR-EMS-Handle}*
OVRHEAPEND	*WORD*	*0*	*{ Pufferende }*
OVRHEAPORG	*WORD*	*0*	*{ Pufferstart}*
OVRHEAPPTR	*WORD*	*0*	*{ Akt. Pufferzeiger }*
OVRHEAPSIZE	*WORD*	*0*	*{ Std. Puffergröße }*
OVRLOADLIST	*WORD*	*0*	*{ geladene Overlays }*

String-Konstanten

String-Konstanten müssen entweder von Apostrophen ('...') oder Anführungszeichen ("...") eingeschlossen werden. Soll ein den String einschließendes Zeichen selbst Bestandteil des Strings sein, so muss es innerhalb des Strings doppelt angegeben werden. z.B.

'a'	*00000061h*
'ba'	*00006261h*
'cba'	*00636261h*
'dcba'	*64636261h*
'a ';	*00006120h*
';a'	*20202061h*
*'a'*2*	*000000E2h*
'a'-'A'	*00000020h*
NOT 'a'	*FFFFFF9Eh*

Typisierte Konstanten

ERRORADDR	*POINTER = NIL*	*{ Laufzeitfehleradresse }*
EXITCODE	*INTEGER = 0*	*{ Exitcode }*
EXITPROC	*POINTER = NIL*	*{ Exit-Prozedur }*
FILEMODE	*BYTE= 2*	*{ Standardmodus für das Öffnen von Dateien }*
HEAPERROR	*POINTER = NIL*	*{ Heap-Fehlerbehandlg }*
HEAPORG	*POINTER = NIL*	*{ Start des Heaps }*
HEAPPTR	*POINTER = NIL*	*{ Akt. Heapobergrenze }*
INOUTRES	*INTEGER = 0*	*{ Status IOResult }*
PREFIXSEG	*WORD= 0*	*{ PSP-Segmentadresse }*
RANDSEED	*LONGINT*	*{ Startwert für den Zufallszahlengenerator }*
STACKLIMIT	*WORD= 0*	*{ Untergrenze Stack }*
TEST8087	*BYTE= 0*	*{ 80x87 vorhanden ? }*

ꕥ Всё ꕥ

Sinnbilder für Informationsverarbeitung

Sinnbilder für Informationsverarbeitung

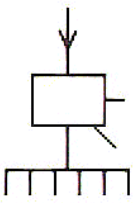
vgl. DIN 66001 (12.83)

Sinnbild	Benennung, Bemerkung
	Verarbeitung, z.B. Addition, Subtraktion Verarbeitungseinheit, z.B. Mensch, Rechner
	Manuelle Verarbeitung, z.B. Lesen, Schreiben Manuelle Verarbeitungsstelle
	Verzweigung, z.B. bei Entscheidung Auswahleinheit, z.B. Schalter
	Schleifenanfang, Beginn eines sich wiederholenden Programmteiles
	Schleifenende, Ende eines sich wiederholenden Programmteiles
	Synchronisierung bei paralleler Verarbeitung; Synchronisiereinheit
	Sprung mit Rückkehr
	Sprung ohne Rückkehr
	Unterbrechung von außen
	Steuerung von außen

Sinnbild	Benennung, Bemerkung
	Daten allgemein Datenträger, allgemein
	Maschinell zu verarbeitende Daten; Datenträger für maschinell zu verarbeitende Daten
	Manuell zu verarbeitende Daten Manuelle Ablage, z.B. Kartei, Archiv
	Daten auf Schriftstück, z.B. Beleg; Ein-/Ausgabeeinheit für Schriftstück, z.B. Belegleser, Drucker
	Daten auf Karte, z.B. Lochkarte Lochkarteneinheit Leser, Stanzer
	Daten auf Lochstreifen Lochstreifeneinheit Leser, Stanzer
	Daten oder Gerät: Speicher mit nur sequentiellem Zugriff, z.B. Magnetband
	Daten oder Gerät: Speicher auch mit direktem Zugriff, z.B. Diskette oder Festplatte

Sinnbild	Benennung, Bemerkung
	Daten im Zentralspeicher Zentralspeicher
	Optische oder akustische Daten, z.B. Bild, Ton; Optische oder akustische Ausgabeeinheit, z.B. Bildschirm, Lautsprecher
	Manuelle optische oder akustische Daten Eingabeeinheit, z.B. Tastatur, Mikrofon
	Verarbeitungsfolge Zugriffsweg
	Datenübertragungsweg
	Grenzstelle zur Umwelt, z.B. Anfang
	Verbindungsstelle, verbindet Darstellungsteile
	Verfeinerung, entspricht Ausschnittvergrößerung
	Bemerkung zur Anfügung erläuternder Texte

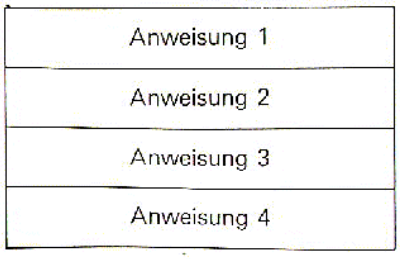
Darstellung von Verbindungslinien



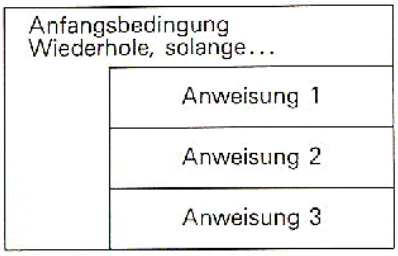
Sinnbilder für Struktogramme (nach Nassi-Shneiderman)

vgl. DIN 66261 (11.85)

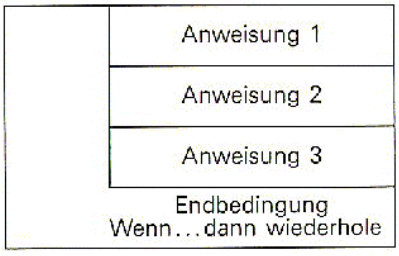
Folgeblock



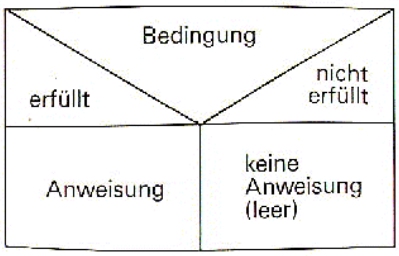
Wiederholungsblock
mit Anfangsbedingung



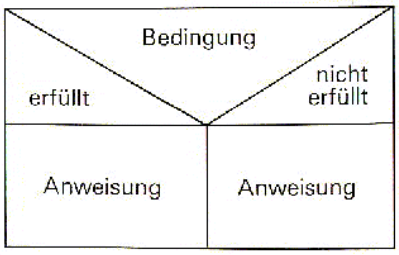
Wiederholungsblock
mit Schlußbedingung



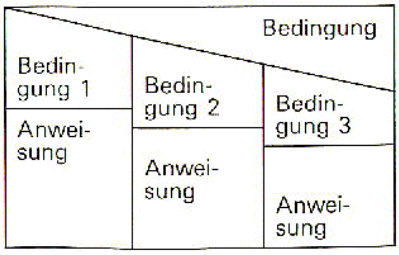
Alternative
Einfache Alternative



Alternative
Bedingte Verarbeitung



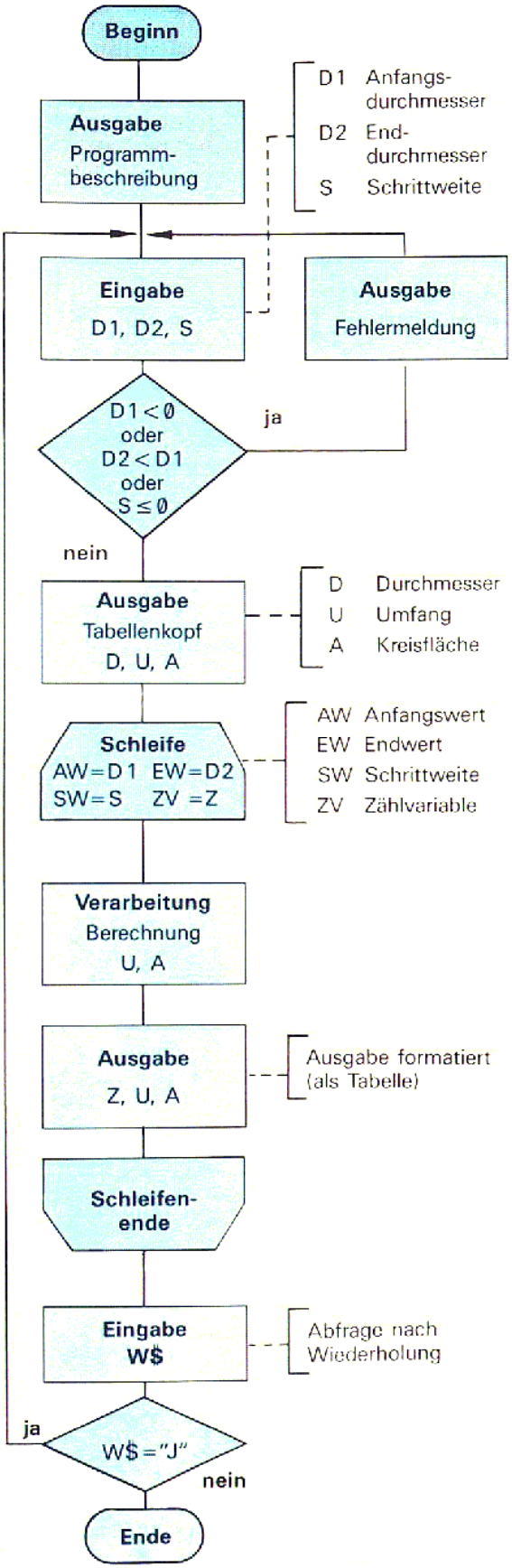
Alternative
Mehrfache Alternative



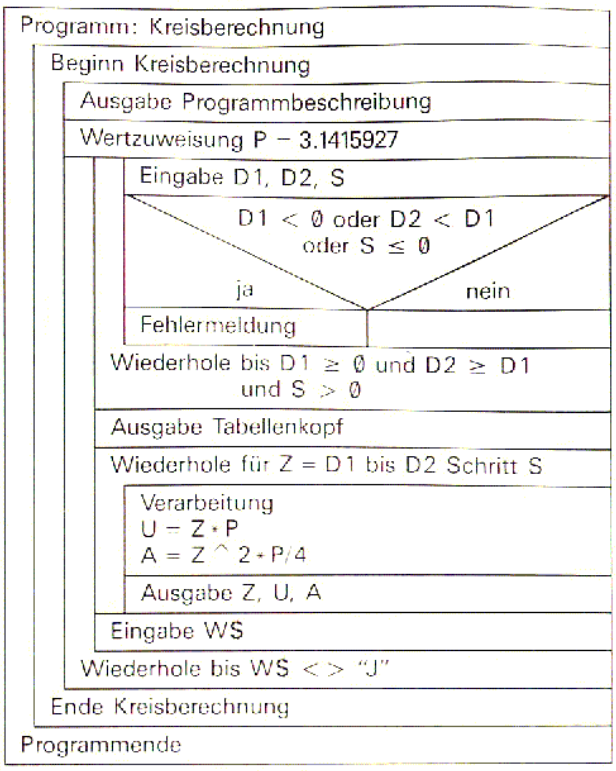
Sinnbilder für Informationsverarbeitung

Programmablaufplan und Struktogramm

Programmablaufplan Beispiel: Kreisberechnung



Struktogramm Beispiel: Kreisberechnung



BASIC-Programm zum Beispiel Kreisberechnung

```
 10 REM    PROGRAMM KREISBERECHNUNG ****
 20 REM    PROGRAMMBESCHREIBUNG
 30 PRINT  "DIESES PROGRAMM BERECHNET"
 40 PRINT  "UMFANG UND FLAECHE VON KREISEN"
 50 PRINT
 60 LET    P = 3.1415927
100 REM    EINGABE VON WERTEN ***********
110 PRINT  "DURCHMESSER-ANFANGSWERT: ";
120 INPUT  D1
130 PRINT  "DURCHMESSER-ENDWERT: ";
140 INPUT  D2
150 PRINT  "SCHRITTWEITE: ";
160 INPUT  S
170 PRINT
200 REM    VERARBEITUNG UND AUSGABE *****
210 IF D1 < 0 THEN 1000
220 IF D2 < D1 THEN 1000
230 IF S < = 0 THEN 1000
240 PRINT  "D", "U", "A"
250 PRINT
260 FOR Z = D1 TO D2 STEP S
270   LET U = Z * P
280   LET A = Z^2 * P/4
290   PRINT Z, U, A
300 NEXT Z
310 PRINT
400 REM    ABSCHLUSS ********************
410 PRINT  "WEITERE BERECHNUNG? (J/N) ";
420 INPUT  W$
430 IF W$ < > "J" THEN END
440 GOTO 100
450 END
1000 REM   FEHLERMELDUNG ***************
1010 PRINT "UNZULAESSIGE EINGABE"
1020 PRINT
1030 GOTO 100
```