

Zusammenstellung wichtiger GWBASIC-Kommandos und -anweisungen:

- Kommandos wirken lediglich im Direktmodus
- Anweisungen können sowohl im Direkt- als auch im Programmmodus (also nacheinander und durch Zeilennummer getrennt) abgearbeitet werden

Die nachfolgenden Kommandos und Anweisungen sind jeweils gross und kursiv geschrieben - ergänzt um die jeweils erforderlichen Parameter und/oder Optionen!

Kommandos:

Beachten Sie, dass immer dann, wenn eine Variable als Parameter angegeben werden kann natürlich auch ein Wert oder eine Konstante gesetzt werden darf!

Nachfolgend in eckigen Klammern (Optionalklammern) geführtes kann entfallen!!!

AUTO[anfangszeile,zeilenabstand]

CONT

- setzt ein mit ^C oder STOP unterbrochenes Programm ab der nächsten Zeilennummer mit den aktuellen (evtl auch geänderten) Variableninhalten fort

COPY zielzeilennummer,zielzeilenabstand=qüllzeilenanfang-qüllzeilenende

- kopiert die durch Qüllzeilenanfang vermerkten Dateien in ihrer Reihenfolge bis Qüllzeilenende nach der Zielzeilennummer in der durch Absatand notierten Numerierung
- die Zielzeilen werden auch neu eröffnet, müssen also noch nicht vorhanden sein

DELETE anfangszeile-endzeile

- löscht den durch "anfangs-" und "endzeile" eingeschlossenen Teil eines BASIC-Qülltextes

FILES["laufwerk:"][\verzeichnis][filter.filter]

- bewirkt die Ausgabe der durch die Filter verbleibenden Dateinamen des angegebenen Laufwerkes und Verzeichnisses

FRE(X\$)

- reorganisiert den Stringspeicher
- gibt die Anzahl der freien Speicherplätze im Stringspeicher

FRE(X)

- reorganisiert den numerischen Variablenspeicher und gibt die Anzahl der freien Bytes zurück
- FREE("") bewirkt nur eine Reorganisation des STRING-Datenspeichers

KEYtastenummer,zeichenfolgenausdruck

- ermöglicht die Neudefinition der Funktionstasten mit bis zu 15 Zeichen
- Konstanten müssen in Anführungszeichen eingeschlossen werden

KEY ON

- Funktionswert der definierten Funktionstasten wird angezeigt

KEY OFF

- Funktionswert der definierten Funktionstasten wird nicht angezeigt

KEY LIST

LIST[zeilennummer]

LLIST[zeilennummer]

LOAD["laufwerk:"][\verzeichnis] dateiname

- lädt eine .BAS Datei in den Arbeitsspeicher, welche durch RUN dann auch gestartet werden kann
- vorhandene Programme mit gleicher Zeilennumerierung werden überschrieben

.LPT1:

MERGE["laufwerk:"][\verzeichnis] dateiname

- wirkt wie LOAD, lediglich wird die so angeforderte Datei an ein schon im Hauptspeicher stehendes Programm angefügt wird
- das durch MERGE angebundene Programm muss im ASCII-Format abgespeichert vorliegen!!!

MKDIR "laufwerk:[\verzeichnis]\neüsverzeichnis"

- legt auf dem spezifizierten Laufwerk und Verzeichnis ein neüs Verzeichnis an
- erfolgt keine Laufwerks- und Verzeichnisangabe, wird das Verzeichnis unterhalb des aktüllen erstellt
- in Anführungszeichen dürfen max. 63 Zeichen stehen

NAME"laufwerk:dateiname" AS "neürdateiname"

- benennt eine Datei um

NEW

- löscht den Arbeitsspeicher

RENUM[neüzeilennummer,altezeilennummer,abstand]

- ordnet die Zeilennummern neu
- beginnend bei Zeile 10 fortlaufend in Zehnerschritten, wenn keine Optionalangaben erfolgten
- GOTO zeilennummer bzw. GOSUB zeilennummer werden in der Rufzeile mit umgeschrieben

RESET

- alle geöffneten Dateien werden geschlossen
- entspricht im Kommandomodus der Anweisung CLOSE ohne Dateinummer

RUN[zeilennummer]

- startet das aktive Programm ab angegebener Zeilennummer

SAVE["laufwerk:"][\verzeichnis] dateiname.bas",a

- zeichnet die angegebene Datei auf entsprechendes Laufwerk und Verzeichnis
- BASIC-Anweisungen werden dabei in TOKEN (Zeichensymbole) umgesetzt
- Nachsatz ,a speichert die Datei im ASCII-Format

SYSTEM

- Rückkehr zum Betriebssystem

TRON

- schaltet die Zeilennummerausgabe in das laufende Programm ein
- dient vor allem der Programmtestung und Fehlersuche
- abgeleitet von "trace on"

TROFF

- schaltet die Zeilennummernaussage im Programmlauf aus
- abgeleitet von "trace off"

Anweisungen:

Viele der Anweisungen lassen sich natürlich auch im Kommandomodus abarbeiten!

Anmerkung 1: Beachten Sie, dass im Textmodus der Monitor von eins beginnend achtzig Zeichen von links nach rechts zulässt und von oben nach unten 25 Zeilen von eins beginnend ermöglicht!
Dies entspricht einem Textmonitor von: 80 mal 25 Zeichen (2000 Zeichen auf dem gesamten Bildschirm).

Anmerkung 2: Im Graphic-modus werden 640 Punkte in x-Richtung (von 0 bis 639 jeweils links nach rechts) sowie 350 in y-Richtung (0 bis 349 von unten nach oben), dies entspricht 224000 ansprechbaren Bildpunkten Bei 16 möglichen Farbcodierungen pro Pixel.
Die hier gemachten Angaben beziehen sich auf CGA-Karten, insbesondere bei HERCULES-Graphikadaptern ist die Simulation mittels Programm erforderlich!!!

vergleichsgrösse1 AND vergleichsgrösse2

- entspricht einer logischen AND-Verknüpfung der beiden Vergleichsgrößen
- das Ergebnis dieser Operation ist WAHR oder FALSCH (TRUE oder FALSE)

numerischevariable=ASC(stringvariable)

- liefert den ASCII-Code des ersten Zeichens der Stringvariable

ATN(numerischevariable)

- ermittelt das Bogenmass des Wertes der numerischen Variable mit einfacher Genauigkeit

BEEP

- realisiert eine Tonausgabe mit fester Frequenz und Zeit

BLOAD"dateiname"[,versatz]

- lädt das Abbild einer Datei in den Speicher
- die Datei darf nicht geöffnet sein
- die Ansprache der Datei erfolgt über die vollständige Pfadangabe (Laufwerk und Verzeichnis)
- "versatz" ist eine numerische Variable im Bereich von 0 bis 65535 und stellt die letzte durch "DEF SEG" vorgenommene Einstellung des Segmentregisters dar

BSAVE"dateiname"[,versatz,länge]

- vor "BSAVE" muss eine "DEF SEG"-Anweisung gegeben werden und damit das Segmentregister des Prozessors eingestellt werden
- speichert das Abbild des Hauptspeichers in die unter "dateiname" hinterlegte Datei
- die nicht bestehende Datei wird geöffnet
- wird keine Länge angegeben, so wird das volle Segment mit 65535 Byte abgelegt

BIN\$

CALL numerischevariable[(argumentenliste)]

- ruft ein Maschinen-Unterprogramm auf
- "numerischevariable" gibt die Startposition des UPs im Speicher an, welches als Versatz in das aktuelle Segment geladen wird

CHR\$(variable)

- realisiert die Übergabe des in Variable enthaltenen ASCII-Codes in eine Stringvariable
- der Inhalt der Variablen darf 255D nicht überschreiten

CIRCLE(xmittelpunkt,ymittelpunkt),farbe,start,ende,achsenverhältnis

- zeichnet Kreise, Ellipsen oder Winkel im Graphicmodus

- x und y geben die Mittelpunktskoordinaten an
- farbe legt die Ausgabefarbe fest (von 0 bis 31)
- "start" und "ende" sind die Winkelangaben
- "achsenverhältnis" legt bei der Ellipse das Verhältnis der beiden Durchmesser fest

CLEAR

- löscht alle Variablen
- Stingvariablen werden leer gesetzt (ihr Wert ist:"")
- numerische Variable erhalten den Wert 0
- darf nicht in Unterprogrammen verwendet werden, da sonst der Rückkehrzeiger mit entfernt wird

CLOSE dateinummer1,dateinummer2

- beendet eine Ein- oder Ausgabeoperation an eine Datei oder ein Gerät

CLS

- löscht den Monitor und setzt den Cursor nach oben links
- PRINT CHR\$(12) bewirkt dies ebenfalls entsprechn ASCII

COLOR hintergrundfarbcodevariable,vordergrundfarbcodevariable

- schaltet die Farbausgabe in die entsprechenden Codes

numerischevariable=COS(variable)

- bestimmt den Cosinus der entsprechenden Variable
- der Variablenwert muss dabei im Bogenmass vorliegen

CONT

- setzt ein mit BREAK unterbrochenes Programm ab der Stelle fort, wo es vorher unterbrochen wurde

CSNG(numerischevariable)

- wandelt eine Variable grösserer Genauigkeit in einfache

numerischevariable=CSRLIN

- liefert die aktuelle Zeilenposition des Cursors im Wert von 1 bis 25
- abgeleitet von CURSOR-LINE

CVD

- wandelt Zeichenfolgenwerte in numerische Werte mit 8 Byte Genauigkeit um
- numerische Werte aus Dateien mit wahlfreiem Zugriff müssen vor ihrer weiteren Verrechnung wieder umgewandelt werden

CVI

- wandelt Zeichenfolgenwerte in numerische Werte mit 2 Byte Genauigkeit um
- siehe CVD

CVS

- wandelt Zeichenfolgenwerte in numerische Werte mit 4 Byte Genauigkeit um
- siehe CVD

DATA wert1, wert2,...,wertn

- hinterlegt Werte, welche durch READ in Variablen eingelesen werden können
- Stringwerte müssen dabei nicht in "string" eingeschlossen werden

stringvariable=DATE\$

- setzt das Datum
- folgende Formate sind gültig: mm-tt-jj
mm/tt/jj
mm-tt-jjjj



mm/tt/jjjj

DATE\$=stringvariable
- übergibt das Datum

DEF FNfunktionsname(argument1,argument2)=mathematischeoperation

DEF USRnummer=versatz
- spezifiziert die Sartadresse eines Unterprogrammes, welches durch USR aus dem Speicher abgerufen wird

DIM variable(1.dimension,2.dimension,...,n.dimension)
- dimensioniert eine Variable als in ihren Ausdehnungen mehrfach
- "versatz" ist die zur augenblicklichen Prozessoradresse bezüglich verschobene Satrtpostion des Unterprogrammes

DRAW"zeichenfolgenausdruck"
- DRAW gilt nur im Graphic-Modus
- zeichnet ein Objekt
- Bewegungsbefehle: Befehl Bewegung

Un	nach oben
Dn	nach unten
Ln	nach links
Rn	nach rechts
En	diagonal nach rechts oben
Fn	diagonal nach rechts unten
Gn	diagonal nach links unten
Hn	diagonal nach links oben

Bewegungspräfixe:	Präfix	Darstellungsart
	B	Bewegung ohne Punkt zu zeichnen
	N	Bewegung und Rückkehr zur Ausgangsposition nach Abschluss der Bewegung
	An	setzt den Winkel "n" mit n=: * 0 = 0 Grad * 1 = 90 Grad * 2 = 180 Grad * 3 = 270 Grad
	TAn	dreht den Winkel "n" im Bereich - 360 Grad bis + 360 Grad * negativ dreht im Uhrzeigersinn * positiv dreht entgegen dem uhrzeigersinn
	Cn	setzt die Farbe "n" in den gültigen Attributen
	Pmalfarbe	spezifiziert Rand- und Füllfarben

EOF
- testet auf die Endekennung einer Datei
- liefert einen Wahrheitswert TRÜ oder FALSE in logischen Tests
- bei Variablenzuweisung ist EOF = -1 bei Ende einer Datei und EOF = 0 bei nichterkanntem Ende
- steht für: END OF FILE

END
- beendet ein Programm ohne die Möglichkeit, mit "CONTINÜ" fortsetzen zu

- können
- der Interpreter kehrt auf die Kommandöben zurück
 - in jedem Programm darf nur eine Anweisung END enthalten sein

ERASE feldvariable1,feldvariable2,feldvariablen

- löscht Felder aus einem Programm

numerischevariable=EXP(X)

- liefert als Ergebnis die Basis des natürlichen Logarithmus hoch "X"
- "X" muss kleiner 88.02969 sein, sonst "Überlauffehlermeldung"

FIELDdateinummer,grösse ASzeichenfolgevariable

- weist Speicherplatz für Variablen im Puffer für wahlfreien Zugriff zu

FOR laufvariable=anfangswert TO endwert[STEP schrittweite]
 anweisungsfolge
 NEXT[laufvariable]

- realisiert den hier als Anweisungsfolge notierten Teil in der Ausführung mehrfach
- die Laufvariable wird ausgehend vom Anfangswert nach jedem Durchlauf um die Schrittweite erhöht
- ist diese im Inhalt gleich der Endvariable, so wird die Schleife abgebrochen
- Standardschleifenvariablen sind in der Reihe: I, J, K, usw.

FIX(variable)

- rundet nicht auf oder ab, sondern schneidet einfach den Nachkommateil auch eines STRINGS ab

GETdateinummer,satznummer

- liest einen Datensatz von einer Datei mit wahlfreiem Zugriff
- "dateinummer" ist die Nummer, unter welcher die Datei geöffnet wurde
- "satznummer" ist die Nummer des gesuchten Datensatzes innerhalb der Datei und von 1 bis 16777215 definiert
- die Datei muss geöffnet sein

GOSUB zeilennummer
 anweisungsfolge
 RETURN

- bewirkt den Aufruf eines Unterprogrammes, welches eine beliebige Anzahl von Zeilennummern enthalten darf und immer mit RETURN abgeschlossen werden muss
- innerhalb eines Unterprogrammes darf keine CLEAR-Anweisung gegeben werden, dies würde die Rücksprungadresse zum rufenden Hauptprogramm abschneiden
- als Unterprogramm werden Programmteile hinterlegt, welche durch einen Algorithmus identisch mehrfach genutzt werden!!!
- veränderliche Parameter können in Variablen hinterlegt werden

GOTO zeilennummer

- bewirkt die unbedingte Verzweigung zur hier angegebenen Zeilennummer
- rückwärts dürfen Sie nur springen, um Schleifen zu schliessen, sonst nur vorwärts
- in sich geschlossene Programmstrukturen sollten nicht durch GOTO verlassen werden, Sie verletzen damit die Regeln der strukturierten Programmierung

stringvariable=HEX\$(X)

- übergibt eine Zeichenfolge die den hexadezimalen Wert des numerischen Arguments darstellt
- "X" wird auf ganze Zahl automatisch aufgerundet und ist von -32768 bis 65535

definiert

INCASES\$=bedingungsstring-variable:eineaktion

- testet einen String auf sein exaktes Vorhandensein und reagiert mit genau einer Aktion
- benötigen Sie mehrere Reaktionen, so verzweigen Sie in ein Unterprogramm

IF bedingung THEN anweisungsfolge für erfüllte bedingung

ELSE anweisungsfolge für nichterfüllte bedingung

stringvariable=INKEY\$

- liest ein Zeichen von der Tastatur ein
- wird kein Code von der Tastatur erkannt, wird die Ergebnisvariable mit einem Leerstring beschrieben

INPUT["text",] variable

- liest einen Inhalt in eine Variable ein
- die Variable muss vom Typ her dem einzulesenden Inhalt entsprechen
- wird mit >ENTER< abgeschlossen

stringvariable=INPUT\$(zeichenzahl,dateinummer)

- liest aus einer geöffneten Datei mit "dateinummer" die entsprechende "zeichenzahl"
- ausser ^C werden alle Codes erkannt und übertragen

numerischevariable=INSTR(verstaz,suchstring,qüllstring)

- sucht das ertmalige Auftreten des "suchstring" "qüllstring" und übergibt die entsprechende Position
- der optional anzugebende Verstz legt den Beginn des Suchens im "qüllstring" fest
- INSTR liefert 0, wenn:
 - versatz > LEN(qülltring\$)
 - der qüllstring Null ist
 - der suchstring nicht gefunden wird

variable=INT(numerischevariable)

- liefert in Variable den ganzzahligen Anteil der Numerischen Variable
- obwohl eine Rundungsfunktion wird bis in die Nähe der nachfolgenden ganzen Zahl nachunten gerundet

KEY ON

- Funktionswert der definierten Funktionstasten wird angezeigt

KEY OFF

- Funktionswert der definierten Funktionstasten wird nicht angezeigt
- sollte an jedem Programmanfang stehen, um die Funktionsatasten für die Laufzeit des Programmes auszublenden

LET variable=variable

- weist einer Variable einen neün Inhalt zu
- die Zuweisung kann auch eine mathematische Operation sein
- zugewiesener Inhalt muss mit dem Typ der Variable übereinstimmen

numerischevariableLEN(stingvariable)

- ermittelt die Länge einer Stringvariablen in Anzahl der erkannten Zeichen

LOC

LOCATE(x-coordinate,y-coordinate)

- setzt den Cursor an die angegebene Position

- vergleiche Anmerkung 1!

LOCK

variable=LOF(dateiname)

- liefert die Länge einer Datei als Anzahl der Bytes
- abgeleitet von: LAST OF FILE

LSET

MKD\$

MKI\$

MKS\$

numerischevariable=variable1 MODvariable2

- ermittelt den Rest der Division Variable 1 : Variable 2

NOT variable

- invertiert (negiert) den Wahrheitswert einer Variablen
- kehrt Binärwerte von Variablen bitweise um

ON variable GOTO zeilennummer1,zeilennummer2,...,zeilennummern

- bewirkt die Verzweigung zu der in der Reihenfolge enthaltenen Zeilennummer, welche in der Variable enthalten sein muss

OPEN

vergleichsgrösse1 OR vergleichsgrösse2

- entspricht einer logischen ODER-Verknüpfung der beiden Vergleichsgrößen
- das Ergebnis dieser Operation ist WAHR oder FALSCH

PALETTE attribut,farbe

- verändert einen oder mehrere Werte der Farbpalette

PALETTE USINGattribut,farbe

- ändert eine ganze Gruppe von Elementen

PAINT(xstart,ystart),malfarbe,randfarbe,hintergrundfarbe

- füllt eine Graphik mit den angegebenen Attributen aus

numerischevariable=PLAY(blindargument)

- liefert die Anzahl der Noten, welche noch im Tonpuffer hinterlegt sind
- 32 ist der grösztmögliche Wert

numerischevariable=POINT(xkoordinate,ykoordinate)

- liest die Koordinaten und Attribute eines Punktes aus und kann so nur auf Wahrheitswert getestet werden
- wird ein Punkt ausserhalb des Bildschirmes gelesen, so ergibt sich -1
- wird statt der Koordinaten ein Funktionswert gesetzt, so gilt:
 - * 0 = aktuelle x-Koordinate
 - * 1 = aktuelle y-Koordinate
 - * 2 = aktuelle x-Koordinate, wenn WINDOW aktiv ist
 - * 3 = aktuelle y-Koordinate, wenn WINDOW aktiv ist

POKE versatzadresseimsegment,byte

- schreibt ein Byte an eine angegebene Speicherstelle
- Adresse und Byte müssen ganzzahlig sein
- Byte ist von 0 bis 255 definiert

- Adressen werden von 0 bis 65535 angegeben
- Hexawerte durch vorgesetztes: "&H" kennzeichnen

numerische variable=POS

- übergibt der Variablen der Inhalt der aktuellen Cursorposition
- linker Bildschirmrand liefert den Wert 1

PRESET xkoordinate,ykoordinate,farbwert

- löscht einen Punkt an angegebener Stelle

PRINT variable

- gibt den Inhalt der Variablen an die aktuelle Cursorposition aus

PRINT AT(x-coordinate,y-coordinate)

- die Ausgabe erfolgt an der vorgegebenen Position
- PRINT allein auf der Zeile erzwingt die Ausgabe einer Leerzeile aus dem aktuell laufenden Programm heraus

PSET xkoordinate,ykoordinate,farbwert

- zeichnet einen Punkt mit eingetragenem Farbwert an festgelegter Stelle- vergleiche Anmerkung 1!

PUT

RANDOMIZE

- startet den Zufallsgenerator

READ variable

- liest den Wert entsprechend der Stellung des DATA-zeigers in die vorgegebene Variable
- Variablen und DATA-Typ müssen übereinstimmen
- der DATA-zeiger wird nach jedem READ automatisch um genau ein erhöht

REM kommentarzeile

- der Inhalt nach der REM-Anweisung wird in der Programmabarbeitung ignoriert

RESTORE[zeilennummer]

- setzt den DATA-zeiger auf den ersten Wert innerhalb der durch die Zeilennummer festgelegten Zeile
- ohne Zeilennummer auf den ersten Wert der ersten DATA-Zeile

RND(ganzzahligekonstante)

- erzeugt eine Zufallszahl zwischen 0 und 1 mit einer Genauigkeit von 9 Nachkommastellen
- um daraus Zahlen grösser 1 zu erhalten wird entsprechend mit Vielfachen von 10 multipliziert
- Konstante sollte 1 sein

RSET

SHELL

- ruft aus BASIC heraus die Betriebssystemebene auf, so dass Manipulationen von MS-DOS heraus ausgeführt werden können
- der Interpreter wird dabei nicht zerstört
- Rückkehr vom Betriebssystem mit "exit" >ENTER<

numerischevariable=SIN(variable)

- bestimmt den Sinuswert der entsprechenden Variable
- der Variablenwert muss dabei im Bogenmass vorliegen

SOUND frequenz,dauer



- erzeugt Töne über den Lautsprecher
- "freqüenz" ist die Tonhöhe in Hertz und im Bereich von 0 bis 32767 gültig
- "daür" gibt die Tonlänge an und ist mit 18,2 pro Sekunde zu verrechnen
- ist "daür" = 0, wird jede aktive SOUND-Anweisung abgeschalten
- 0>"daür"<0,022 ergibt einen unendlich langen Ton
- Tonhöhen: Note Freqüenz Note Freqüenz

C	130.810	C	523.250
D	146.830	D	587.330
E	164.810	E	659.260
F	174.610	F	698.460
G	196.000	G	783.990
A	220.000	A	880.000
B	246.940	B	987.770
C	261.630	C	1046.500
D	293.660	D	1174.700
E	329.630	E	1318.500
F	349.230	F	1396.900
G	392.000	G	1568.000
A	440.000	A	1760.000
B	493.880	B	1975.500

- SOUND 32767,daür liefert keinen Ton!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SQR(argument)

- Berechnung der Quadratwurzel des Arguments

numerischevariable=STICK(koordinate)

- liefert als Ergebnis die x- und y-Koordinaten von zwei Joysticks
- "koordinate" kann die Werte 0 bis 3 annehmen, wobei gilt:
- * 0 = x-Koordinate von Joystick A
- * 1 = y-Koordinate von Joystick A
- * 2 = x-Koordinate von Joystick B
- * 3 = y-Koordinate von Joystick B

STOP

- bewirkt das Anhalten des Programmlaufes an dieser Stelle und Rückkehr in den Kommandomodus

TAB(zeichenzahl)

- schaltet die Ausgabe um den Wert Zeichenzahl horizontal nach rechts

numerischevariable=TAN(variable)

- bestimmt den Tangenswert der entsprechenden Variable
- der Variablenwert muss dabei im Bogenmass vorliegen
- der Cotangens ergibt sich aus SIN(x) : COS(x)

stringvariable=TIME\$

- Übergabe der aktuellen Uhrzeit in eine Stringvariable

numerischevariable=TIMER

- übergibt in tausendstel zu messende Sekundenwerte an die Variable
- ist vor allem in seiner Differenz zur Programmlaufzeitberechnung interessant

TIME\$"HH:SS"

- setzt für die Laufzeit des aktuellen Programmes die Zeit
- diese wird nicht auf der Betriebssystemebene manipuliert

TIMER ON

- aktiviert den TIMER für ON TIMER(Sec) GOSUB zeilennummer

TIMER OFF

- schaltet den TIMER entsprechend TIMER ON ab (im Sinne von abbrechen)

TIMER STOP

- hält die TIMER-Ausführung lediglich an (im Sinne von unterbrechen)

UNLOCK

VAL(stringvariable)

- wird dann genutzt, wenn in eine Strinvariable ein numerischer Teile eingelesen wurde und dieser verrechnet werden soll
- mit STRINGS können Sie nicht rechnen

WIDTH[LPRINT]breite

WIDTHdateneinrichtung

WIDTHgerät,breite

- es wird eine Druckzeilenbreite an die Ausgabeeinheit übermittelt

* SCRN: Bildschirm

* LPT1: 1. Drucker

* LPT2: 2. Drucker

* LPT3: 3. Drucker

* COM1: 1. Datenfernübertragungsanschluss

* COM2: 2. Datenfernübertragungsanschluss

WHILE bedingung

anweisungsfolge

WEND

- wiederholt eine Anweisungsfolge, solange die Bedingung erfüllt ist

WINDOW[[screen](xanfang,yanfang)-(xende,yende)]

- legt die Grösse des grafischen Ausgabefensters fest
- alle Anfangswerte müssen kleiner als die Endwerte sein
- x kann im Bereich von 0 - 639,
- y kann im Bereich von 0 - 349 liegen
- wird der Zusatz "SCREEN" eingefügt, so wird das karthesische Koordinatensystem invertiert dargestellt