

G r a f i k
h a n d b u c h

Für **SHARP**

C o m p u t e r



Klaus Schreiner
Fischel GmbH
Do not sale!

ISBN 3-924327-04-1

G r a f i k
h a n d b u c h

Für **SHARP**

C o m p u t e r



Klaus Schreiner
Fischel GmbH

Do not sale !
ISBN 3-924327-04-1

Klaus Schreiner (Autor)

Fischel GmbH (Hrsg.):

Grafik-Handbuch für Sharp CE-150, CE-515P und CE-516P

Berlin, 1985.

ISBN 3-924327-04-1

© 1985 Fischel GmbH, Berlin.

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem (Fotokopie, Mikrokopie) oder sonstigem Wege zu vervielfältigen.

Für etwaige Schäden durch Anwendung der Anleitungen oder Programme dieses Buches übernehmen wir keine Haftung.

SHARP-COMPUTER

FISCHEL GMBH

KAISER-FRIEDRICH-STRASSE 54A

1000 BERLIN 12

Te1. 030 / 323 60 29

Druck: Offsetdruckerei Gerhard Weinert

Friedrichstraße 224, 1000 Berlin 61

Do not sale !

INHALTSVERZEICHNIS

1	Einleitung	7
2	Die Vierfarbplotter CE-150, CE-515P und CE-516P	8
2.1	Funktionsweise eines Plotters	8
2.2	Anschluß der Plotter an Sharp-Computer	10
2.3	Elementare Grafik-Befehle	10
2.4	Einführende Beispiele	18
3	Grafik-Grundaufgaben und Grafik-Editor	23
3.1	Grafik-Grundaufgaben	25
3.1.1	Verschieben des Zeichensstiftes	25
3.1.2	Zeichnen von Linienzügen und Vielecken	26
3.1.3	Zeichnen von Rechtecken: Teil 1	29
3.1.4	Zweidimensionale Koordinatentransformation	30
3.1.5	Zeichnen von Rechtecken: Teil 2	36
3.1.6	Zeichnen von Quadraten	37
3.1.7	Zeichnen von Senkrechten zu Linien	38
3.1.8	Zeichnen von Dreiecken	40
3.1.9	Zeichnen von Kreisen und Kreisbogen: Teil 1	41
3.1.10	Zeichnen von Kreisen: Teil 2	44
3.1.11	Zeichnen von Kreisbogen: Teil 2	48
3.1.12	Zeichnen von Ellipsen	50
3.1.13	Zeichen von Sonderzeichen	52
3.2	Der Grafik-Editor GRADOR	55
4	Kreisdiagramme	80
5	Balkendiagramme	89
6	Zeichnen von Funktionen	96
7	Schraffur von Vielecken	108
8	Computeranimation	119

Do not sale !

9	Grundlagen der 3D - Grafik	130
9.1	Projektionsformeln der 3D - Grafik	130
9.2	Das Hidden-Line-Problem	136
9.2.1	Der Flächennormalen-Test	136
9.2.2	Der Linien-Flächen-Test	139
9.3	Ein Hidden-Line-Programm	140
10	Zusammenfassung	154

Do not sale !

1 EINLEITUNG

In den letzten Jahren wurde für viele Computer-Anwendungen der Bedarf an grafischen Darstellungen immer größer. So ist es oft anschaulicher, Ergebnisse in Form von Bildern als in Form von Tabellen auszuwerten. Die Fa. Sharp bietet mit den Vierfarbplotttern CE-150, CE-515P und CE-516P leistungsfähige Geräte an, die den Wunsch nach spezieller Grafiksoftware aufkommen lassen.

In diesem Buch werden die Grundlagen der Computer-Grafik in Theorie und in Programmen erläutert. Die vorgestellten Programme wollen aber keine Superprogramme sein. Vielmehr soll mit ihnen der Leser in die Lage versetzt werden, eigene Programme für seine speziellen Bedürfnisse zu erstellen. Natürlich kommen auch diejenigen nicht zu kurz, die sich mit dem Buch nur über die Computer-Grafik informieren wollen. Für diesen Personenkreis findet sich nach jedem Abschnitt ein ausführliches Literaturverzeichnis, in dem Informationen über weiterführende Theorie und über veröffentlichte Programmlistings zu finden sind. Weiterhin werden fertige und auf Kassetten erhältliche Programme kurz vorgestellt.

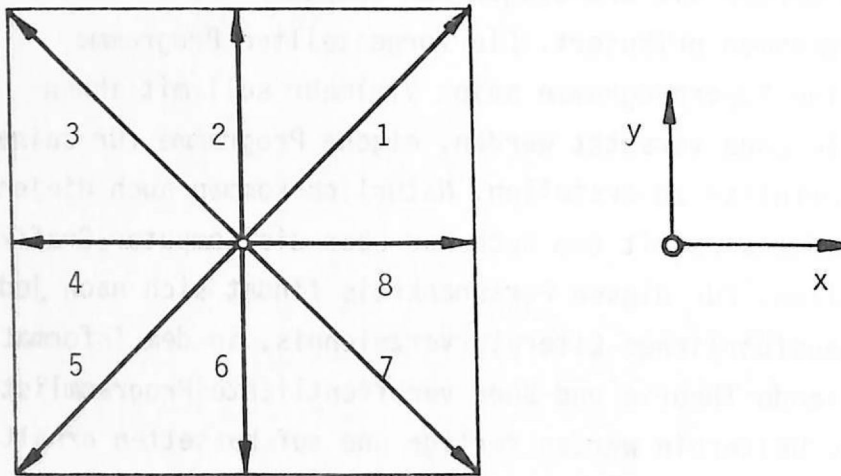
Noch ein Hinweis: Die in diesem Buch abgedruckten Programme sind für den PC-1500 (A) erstellt worden. Sie sind so allgemein und nach Möglichkeit rechnerunspezifisch formuliert, daß sie von Lesern mit anderen BASIC-Rechnern ohne große Probleme angewendet werden können. Grundkenntnisse in BASIC sind allerdings für das Verständnis des Buches Voraussetzung. Dagegen wurde zu Gunsten einer breiten Leserschaft dort, wo es möglich ist, auf komplizierte mathematische Formeln verzichtet. Der Geometriespezialist möge mir deswegen so manche unmathematische Formulierung verzeihen.

Do not sale !

2 DIE VIERFARBPLOTTER CE-150 UND CE-515P / 516P

2.1 FUNKTIONSWEISE EINES PLOTTERS

Ein Plotter ist ein eigentlich sehr einfaches Gerät. Er kennt im Prinzip nur acht verschiedene Fahrbefehle und die beiden Sonderbefehle "Stift heben" und "Stift senken". Die Fahrbefehle bewirken einen Schritt von einer Einheit in Richtung der positiven oder negativen x- bzw. y-Achse.

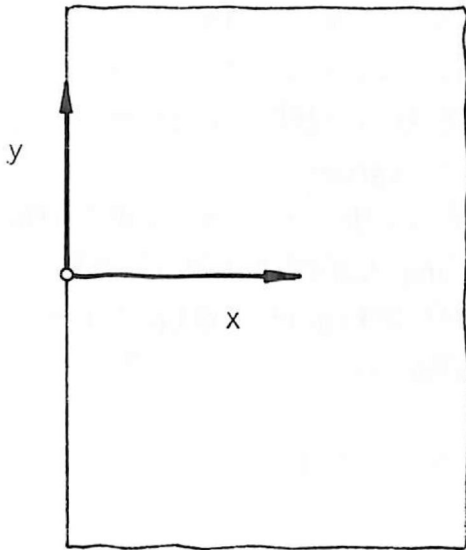


Die Fahrbefehle 1, 3, 5 und 7 resultieren aus einem gleichzeitigen Schritt in x- und y-Richtung. Bei den hier betrachteten Plottern entspricht eine Einheit $\varnothing.2$ mm. Die Fahrt mit gesenktem Stift zeichnet eine Linie, die Fahrt mit gehobenem Stift verschiebt den Stift ohne zu zeichnen. Dabei wird die Bewegung in x-Richtung durch die Stiftbewegung, die Bewegung in y-Richtung durch eine Verschiebung des Papiers bewirkt. Alle grafischen Darstellungen werden durch wiederholtes Anwenden dieser Befehle erstellt.

Natürlich stellt der Computer Hilfsmittel zur Verfügung. Zum Beispiel braucht man Buchstaben und Ziffern nicht umständlich durch mehrfachen Aufruf der Grundbefehle darzustellen. Der BASIC-Rechner

Do not sale !

unterstützt durch eingebaute Programme solche Aufgaben. Auch längere Fahrbefehle über mehrere Einheiten hinweg sind programmiert. Dazu benötigt der Rechner die Angabe des Zielpunktes. Dieser wird durch seine Koordinaten beschreiben. Aus diesem Grund haben die Plotter ein internes Koordinatensystem:



Beim Einschalten des Plotters steht der Zeichenstift am linken Rand. Hier ist der Ursprung, der Nullpunkt des Koordinatensystems. Die x-Achse geht nach rechts, die y-Achse nach oben. Die Skalierung der Achsen erfolgt in 0.2 mm - Schritten. Die folgende Tabelle gibt Auskunft über die maximale Formatbreite in x-Richtung. Die Bewegung in y-Richtung ist im Prinzip nur durch die Länge des Zeichenpapiers bestimmt.

	CE-150	CE-515P / 516P
max. Formatbreite in x-Richtung	220 Einheiten = 44 mm	960 Einheiten = 192 mm

Ein Punkt $x = 150$, $y = 40$ liegt also auf dem Papier an der Stelle $x = 150 * 0.2 \text{ mm} = 30 \text{ mm}$, $y = 40 * 0.2 \text{ mm} = 8 \text{ mm}$.

Do not sale !

2.2 ANSCHLUSS DER PLOTTER AN SHARP-COMPUTER

Über den Anschluß der Plotter CE-150 und CE-515P / 516P geben die entsprechenden Bedienungsanleitungen genaue Auskunft. Der CE-150 läßt sich nur an den PC-1500 (A) anschließen, die CE-515P / 516P - Plotter über eine RS-232C- oder eine Parallel-Schnittstelle an beliebige Rechner. Bei Verwendung der CE-515P / 516P - Plotter am PC-1500 (A) muß die Schnittstelle CE-158 verwendet werden. Diese muß zu Beginn initialisiert werden, wie es in der Bedienungsanleitung zum CE-158 oder noch besser in der Anleitung zum CE-515 P / 516P beschrieben ist. Bei Verwendung des CE-515L - Kabels ergeben sich folgende vier BASIC-Initialisierungskommandos:

```
SETCOM 1200,8,N,1
SETDEV P0
OUTSTAT 0
CONSOLE 0,0
```

Dazu müssen die DIP-Schalter der Plotter folgendermaßen stehen:

Schalter 2 und 6 auf OFF, alle anderen auf ON

Beim Anschluß der CE-515P / 516P - Plotter an andere Rechner ist gemäß der dazugehörenden Bedienungsanleitung vorzugehen.

2.3 ELEMENTARE GRAFIK-BEFEHLE

Im folgenden werden die wichtigsten Grafik-Befehle vorgestellt. Dabei werden der Befehlssatz des CE-150 und der des CE-515P / 516P parallel verwendet und einander gegenübergestellt. Bitte betrachten Sie aber die folgenden Seiten nicht als Ersatz für die Bedienungsanleitungen der einzelnen Drucker. Hier werden nur die wichtigsten

Do not sale !

Merkmale der einzelnen Befehle aufgezeigt.

Zweck der Aufstellung ist es, dem Benutzer die "Übersetzung" der CE-150-Befehle in CE-515P / 516P - Befehle und umgekehrt zu ermöglichen. Damit wird das Ändern selbst erstellter oder erworbener Grafik-Programme erleichtert.

Während die CE-150-Grafik-Befehle vom Namen und von der Syntax her sehr einfach sind, ist dies bei den entsprechenden CE-515P / 516P - Befehlen anders. Das kommt daher, daß der CE-515P / 516P - Plotter über ein Interface angesteuert wird. Dieses Interface versteht im wesentlichen nur das Kommando LPRINT. Alle Grafik-Befehle sind deswegen mehr oder weniger umständliche LPRINT - Konstruktionen.

Beginnen wir also mit der Gegenüberstellung der Grafik-Befehle.

GRAPH	LPRINT CHR\$(27) + "b"
-------	------------------------

Um die Grafik-Befehle verwenden zu können, muß zuerst der GRAPH-Modus des Druckers eingeschaltet werden. Die Rückkehr in den TEXT-Modus erfolgt mit

TEXT	LPRINT CHR\$(27) + "a" oder LPRINT "A"
------	---

Der LPRINT "A" - Befehl funktioniert nur, wenn man sich gerade im GRAPH-Modus befindet.

Mit

GLCURSOR (100,200)	LPRINT "M 100, 200"
--------------------	---------------------

Do not sale !

wird der Zeichenstift von der aktuellen Stelle aus an den Punkt $x = 100$, $y = 200$ bewegt. Dabei ist der Zeichenstift oben; es wird keine Linie gezeichnet.

Wichtig: Alle Vergleiche sind Beispiele! Die allgemeine Syntax der Befehle ist den Bedienungsanleitungen zu entnehmen.

Kompliziert wird es für den CE-515P / 516P, wenn die Koordinaten des Zielpunktes in Variablen stehen:

	$X = 100 : Y = 200$
GLCURSOR (X,Y)	LPRINT "M";X;"",";Y

Das Zeichnen von Linien geschieht mit

LINE - (50,100)	LPRINT "D 50, 100"
-----------------	--------------------

oder

	$X = 50 : Y = 100$
LINE - (X,Y)	LPRINT "D";X;"",";Y

Diese Linien werden von dem Punkt aus gezeichnet, wo der Stift gerade steht. Umständlich ist auch das Zeichnen von Linien von einer Stelle aus, wo der Stift nicht gerade steht:

LINE (0,10) - (20,50)	LPRINT "M 0, 10" LPRINT "D 20, 50"
-----------------------	---------------------------------------

Do not sale !

Hier muß beim CE-515P / 516P zuerst der Stift an den Linienanfang gefahren werden.

Oder:

X1 = 0 : Y1 = 10	
X2 = 20 : Y2 = 50	
LINE (X1,Y1) - (X2,Y2)	LPRINT "M";X1;",";Y1
	LPRINT "D";X2;",";Y2

Längere Linienzüge werden entsprechend eingegeben:

X1 = 0 : Y1 = 0	
X2 = 10 : Y2 = 10	
X3 = 20 : Y3 = 90	
LINE (X1,Y1) - (X2,Y2)	LPRINT "M";X1;",";Y1
- (X3,Y3)	LPRINT "D";X2;",";Y2
	LPRINT "D";X3;",";Y3

Den Linientyp (durchgezogen, gepunktet) gibt man beim CE-150 in das LINE - Kommando ein. Beim CE-515P / 516 P wird der Linientyp separat vorgegeben:

	L = 2
LINE - (100,50),L	LPRINT "L";L
	LPRINT "D 100, 50"

Die Zahl der möglichen Linienarten ist bei den beiden Plottertypen unterschiedlich:

Do not sale !

CE-150 : L = 0 ... 9 (L = 9 : Stift gehoben)
CE-515P / 516P : L = 0 ... 15

Es gibt auch die Möglichkeit der Eingabe von relativen Linienkoordinaten:

	L = 2
RLINE (100,50) - (200,10),L	LPRINT "L";L LPRINT "M 100, 50" LPRINT "J 200, 10"

Den RLINE-Befehl mit L = 9 des CE-150 (relative Bewegung mit gehobenem Stift) gibt es beim CE-515P / 516P als eigenen Befehl:

RLINE -(10,20),9	LPRINT "R 10, 20"
------------------	-------------------

Mit dem SORGN-Befehl kann man den Ursprung des Koordinatensystems neu definieren:

SORGN	LPRINT "I"
-------	------------

Der Koordinatennullpunkt liegt dann an der Stelle, an der sich der Stift gerade befindet.

Mit

GLCURSOR (0,0)	LPRINT "H"
----------------	------------

kehrt der Zeichenstift an den Ursprung des Koordinatensystems zurück.

Do not sale !

Die Farbe wird gewählt mit

COLOR 2	LPRINT CHR\$(27) + "2"
---------	------------------------

Oder:

COLOR C	C = 3 LPRINT CHR\$(27) + CHR\$(&30+C)
---------	--

Nicht nur im TEXT-Modus, sondern auch im GRAPH-Modus kann man Texte schreiben:

LPRINT "Test"	LPRINT "PTest"
---------------	----------------

Oder:

LPRINT A\$	A\$ = "Test" LPRINT "P";A\$
------------	--------------------------------

Die Zeichengröße wird mit

CSIZE 1	LPRINT CHR\$(27) + "?" + "a"
---------	------------------------------

oder

CSIZE 3	LPRINT CHR\$(27) + "?" + "c"
---------	------------------------------

Do not sale !

oder

CSIZE S	S = 3 LPRINT CHR\$(27) + "?" + CHR\$(&60+S)
---------	---

gewählt.

Beim CE-150 geht die Zeichengröße von 1 bis 9, beim CE-515P von 1 bis 15 und beim CE-516P von 1 bis 63.

Analog zum ROTATE-Befehl kann man auch beim CE-515P / 516P die Schreibrichtung wählen:

ROTATE R	R = 2 LPRINT "Q";R
----------	-----------------------

Damit sind alle möglichen Befehle des GRAPH-Modus beim CE-150 und CE-515P aufgeführt. Der CE-516P verfügt über einige weitere Grafik-Befehle. Er kann Kreise und Ellipsen zeichnen und er kann Kreise, Ellipsen und Vielecke schraffieren. Weiterhin kann man den Zeichen- und Zeilenabstand vorgeben. Auch das Zeichnen von Koordinatenachsen ist möglich. Wie diese Befehle funktionieren, entnehme man bitte der Bedienungsanleitung zum CE-516P. Es hat keinen Sinn, diese Befehle dem CE-150 oder CE-515P gegenüberstellen zu wollen, da diese beiden Plotter keine vergleichbaren Befehle haben. Bei diesen Geräten muß man selbst Programme zum Kreise- und Ellipsenzeichnen und zur Schraffur schreiben (vergl. Abs. 3.1.9 ... 3.1.12 und Abs. 7).

Im TEXT-Modus sind die Unterschiede zwischen den Druckern nicht groß. Das LPRINT - Kommando zum Schreiben von Texten ist bei den Plottern gleich. Farbwechsel und Ändern der Zeichengröße erfolgen wie im GRAPH-Modus. Beim Zeilenvorschub

Do not sale !

LF 1 LF -1	LPRINT CHR\$(10) LPRINT CHR\$(11)
---------------	--------------------------------------

gibt es beim CE-515P / 516P immer nur die Änderung um eine Zeile.

Nur beim CE-515P / 516P ist der Zeichen-Rückwärtsschritt möglich:

---	LPRINT CHR\$(8)
-----	-----------------

Mit dieser Gegenüberstellung der Grafik-Befehle von CE-150 und CE-515P / 516P müßte es jedem Leser möglich sein, Programme, die für einen Plotter erstellt sind, für die Verwendung auf einem anderen Plotter umzuschreiben.

Bei den folgenden Programmlistings ist die Verwendung beider Plotter-typen vorgesehen. Durch die BASIC-Abfrage

```
INPUT "Externer Drucker ? (J/N) ";D$  
IF D$ <> "J" LET D$ = ""
```

wird der jeweilige Drucker ausgewählt. (Externer Drucker : CE-515P / 516P.) Leser, die nur einen der Drucker besitzen, können die für sie nicht interessanten Programmzeilen weglassen.

ACHTUNG: Bei der Angabe von Zeichnungskordinaten beim CE-515P / 516P hatte ich Schwierigkeiten. Wenn man z.B. eine Linie zum Punkt (X,Y) hin zeichnen will

```
LPRINT "D";X;",";Y
```

muß man zwei Dinge beachten, die in der Bedienungsanleitung nicht erwähnt sind:

Do not sale !

1. Vor dem LPRINT-Kommando darf kein USING ... verwendet werden, sonst kann es zum Fehlverhalten des Plotters führen. Deswegen sollte man vor dem Kommando LPRINT prinzipiell die USING-Formatierung mit USING ohne Zusatz aufheben.
2. Der Plotter CE-515P / 516P akzeptiert als Koordinaten nur Integer-Größen (Zahlen ohne Kommastellen). Normalerweise ignoriert der Plotter die Nachkommastellen ohne Probleme: $X = 11.53$ wird also zu $X = 11$. Wenn der Inhalt der Koordinatenvariablen aber sehr klein wird, kann es ebenfalls zum Fehlverhalten führen. Ich hatte nach Koordinatentransformationen bedingt durch Rundungsfehler des Rechners Koordinatenangaben wie $X = 1.E-10$ statt $X = 0$. Solche Zahlen verkraftet der Plotter nicht! Zur Abhilfe sollte man die Koordinaten immer in Integer-Größen verwandeln:

```
LPRINT "D";INT X;",";INT Y
```

2.4 EINFÜHRENDE BEISPIELE

Ohne zu sehr ins Detail gehen zu wollen, werden zwei kleine Programme zum Zeichnen von Figuren gezeigt. Diese einführenden Beispiele sollen beim Leser Interesse für die weiteren Abschnitte des Buches wecken und gleichzeitig deutlich machen, mit welchen einfachen Mitteln man Grafik-Programme erstellen kann. Wenn wir dann am Ende des Buches angelangt sind, mag jeder selbst versuchen, mit den gewonnenen Kenntnissen die beiden Programme zu analysieren.

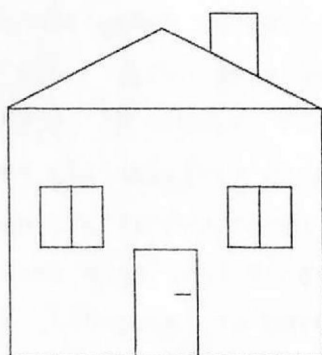
Noch ein Hinweis zur zweiten Figur: Wer genau hinschaut, erkennt, daß die Figur nur aus Quadraten besteht.

Do not sale !

```
1000 "H"
1010 REM -----
1020 REM  BEISPIEL : HAUS
1030 REM -----
1040 INPUT "Externer Drucker ? (J/N) ";D$
1050 IF D$<>"J"LET D$=""
1060 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PO
1070 IF D$<>" "CONSOLE 0,0
1080 IF D$=""GRAPH :GLCURSOR (0,-250):SORGN
1090 IF D$<>" "LPRINT CHR$(27)+"b":LPRINT "M0,-300"
1100 IF D$<>" "LPRINT "I"
1110 RESTORE "HAUS"
1120 "H-NEXT"
1130 READ J
1140 IF J=-999GOTO "H-END"
1150 READ U,V
1160 IF D$=""GLCURSOR (U,V)
1170 IF D$<>" "LPRINT "M";U;" ";U
1180 FOR I=1TO J
1190 READ U,V
1200 IF D$=""LINE -(U,V)
1210 IF D$<>" "LPRINT "D";U;" ";U
1220 NEXT I
1230 GOTO "H-NEXT"
1240 "H-END"
1250 IF D$=""GLCURSOR (0,0):TEXT
1260 IF D$<>" "LPRINT "H":LPRINT "A":SETDEV
1270 END
1280 "HAUS"
1290 DATA 5
1300 DATA 0,0
1310 DATA 200,0
1320 DATA 200,160
1330 DATA 100,210
1340 DATA 0,160
1350 DATA 0,0
1360 DATA 3
1370 DATA 80,0
1380 DATA 80,70
1390 DATA 120,70
1400 DATA 120,0
1410 DATA 1
1420 DATA 105,40
1430 DATA 115,40
1440 DATA 1
```

Do not sale !

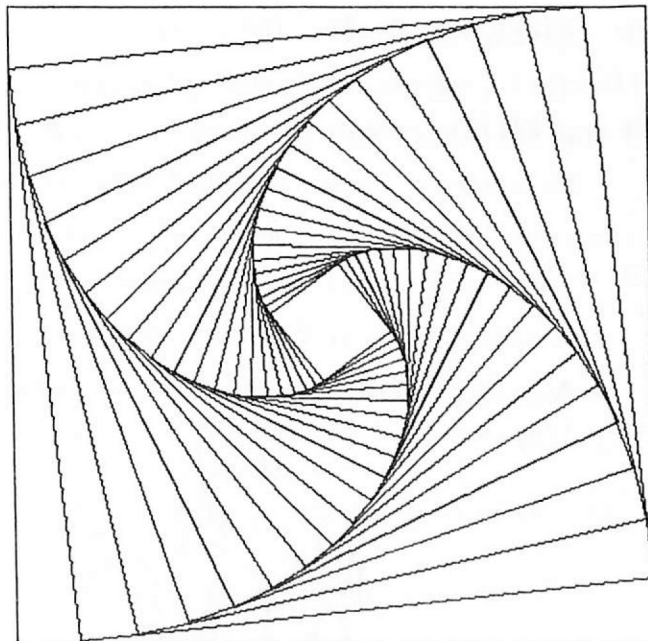
```
1450 DATA 0,160
1460 DATA 200,160
1470 DATA 3
1480 DATA 130,195
1490 DATA 130,220
1500 DATA 160,220
1510 DATA 160,180
1520 DATA 1
1530 DATA 40,110
1540 DATA 40,70
1550 DATA 4
1560 DATA 20,70
1570 DATA 60,70
1580 DATA 60,110
1590 DATA 20,110
1600 DATA 20,70
1610 DATA 1
1620 DATA 160,110
1630 DATA 160,70
1640 DATA 4
1650 DATA 140,70
1660 DATA 180,70
1670 DATA 180,110
1680 DATA 140,110
1690 DATA 140,70
1700 DATA -999
```



```
2000 "Q"
2010 REM -----
2020 REM BEISPIEL : QUADRATE
2030 REM -----
2040 DIM U(3),V(3),X(3),Y(3)
2050 INPUT "Externen Drucker ? (J/N) ";D$
```

Do not sale !

```
2060 IF D$<>"J"LET D$=""
2070 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PO
2080 IF D$<>" "CONSOLE 0,0
2090 IF D$=""GRAPH :GLCURSOR (100,-100):SORGN
2100 IF D$<>" "LPRINT CHR$(27)+"b":LPRINT "M200,-200"
2110 IF D$<>" "LPRINT "I":LPRINT "L0"
2120 U(0)=-100:U(1)=100:U(2)=100:U(3)=-100
2130 U(0)=-100:U(1)=-100:U(2)=100:U(3)=100
2140 IF D$<>" "FOR I=0TO 3:U(I)=U(I)*2:U(I)=U(I)*2:NEXT I
2150 M=.1:N=1-M
2160 FOR I=0TO 20
2170 IF D$=""LINE (U(0),U(0))-(U(1),U(1))-(U(2),U(2))
2180 IF D$=""LINE -(U(3),U(3))-(U(0),U(0))
2190 IF D$<>" "LPRINT "M";U(0);", ";U(0)
2200 IF D$<>" "LPRINT "D";U(1);", ";U(1);", ";U(2);", ";U(2)
2210 IF D$<>" "LPRINT "D";U(3);", ";U(3);", ";U(0);", ";U(0)
2220 FOR J=0TO 3
2230 J1=J+1
2240 IF J=3LET J1=0
2250 X(J)=N*U(J)+M*U(J1)
2260 Y(J)=N*U(J)+M*U(J1)
2270 NEXT J
2280 FOR J=0TO 3
2290 U(J)=X(J)
2300 U(J)=Y(J)
2310 NEXT J
2320 NEXT I
2330 IF D$=""GLCURSOR (0,-100):TEXT
2340 IF D$<>" "LPRINT "M0,-200":LPRINT "A":SETDEV
2350 END
```



Do not sale !

LITARATUR:

- : Sharp Taschencomputer PC-1500, Bedienungsanleitung
- : Sharp Farb-Plotter/Drucker CE-515P, Bedienungsanleitung
- : Sharp Farb-Plotter/Drucker CE-516P, Bedienungsanleitung
- : Sharp Schnittstelle für PC-1500: CE-158, Bedienungsanleitung
- : POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung.
1. Aufl.; Würzburg 1984
- : ANGELL, Ian O.: A Practical Introduction to Computer
Graphics. London 1981

PROGRAMM-LISTINGS FÜR PC-1500 (A):

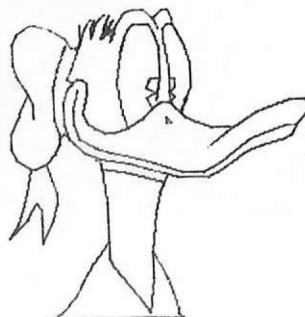
- : Karikaturen per Computer. Happy Computer 5/1984
(Bildeingabe durch Eingabe von Punktkoordinaten)

PROGRAMM-KASSETTEN FÜR PC-1500 (A):

(Erhältlich bei der Fischei GmbH, Berlin)

: BILDKONSTRUKTION und BILDZEICHNER

(Bildeingabe durch Eingabe von Punktkoordinaten. Möglichkeit der Speicherung von Bildern auf Kassette. Möglichkeit der Bildmanipulation. Mit der Kassette werden als Beispiele die Daten für folgende Bilder geliefert:



Do not sale !

3 GRAFIK-GRUNDAUFGABEN UND GRAFIK-EDITOR

Mit den im Abschnitt 2.3 vorgestellten Grafik-Befehlen lassen sich alle denkbaren Grafik-Aufgaben erfüllen. Allerdings ist das Eingeben von Punktkoordinaten auf Dauer eine lästige Angelegenheit.

Es wird deswegen in diesem Abschnitt ein sogenannter "Grafik-Editor" vorgestellt. Ein Grafik-Editor ist ein Programm, das mit Hilfe von implementierten Grafik-Aufgaben das Zeichnen von Bildern erleichtert. Die Bilder können bei Bedarf geändert, neu gezeichnet und auf Kassette gespeichert werden.

Bevor nun der Grafik-Editor im einzelnen vorgestellt wird, sollen die in ihm enthaltenen Lösungen für die wichtigsten Grafik-Grundaufgaben vorgestellt werden. Dabei wurde nach Möglichkeit bei den einzelnen Teilprogrammen die Zeilennummerierung des Grafik-Editors beibehalten. Dadurch wird es erleichtert, die Grafik-Aufgaben im Grafik-Editor wiederzufinden.

Alle Programmbeispiele sind für die Verwendung beider Plotter-typen (CE-150 und CE-515P / 516P) vorgesehen. Deswegen muß zu Beginn der Plottertyp bestimmt werden. Ebenfalls wird vorausgesetzt, daß der Grafik-Modus eingeschaltet wurde. Alle Winkelangaben sind in Alt-Grad.

```
10 WAIT 0:INPUT "Externer Drucker ? (J/N) ";D$
20 IF D$<>"J"LET D$=""
30 IF D$=""GRAPH
40 IF D$<>""SETCOM 1200,8,N,1:SETDEV PO:CONSOLE 0,0:
OUTSTAT 0
50 IF D$<>""LPRINT CHR$ (27)+"b"
60 DEGREE
```

Do not sale !

Allen Beispielen gemeinsam ist folgende Belegung von Variablen:

C :	Nummer der Farbe	(COLOR)
L :	Nummer der Linienart	(LINE)
R :	Nummer der Schreibrichtung	(ROTATE)
S :	Nummer der Zeichengröße	(CSIZE)

Die Koordinaten der Zeichung werden nicht mit X und Y, sondern mit U und V bezeichnet. Das geschieht, um zwischen den Koordinaten eines Gegenstandes im Raum (X, Y, Z) und den Koordinaten der Gegenstandsprojektion auf die Zeichenebene (U, V) unterscheiden zu können.

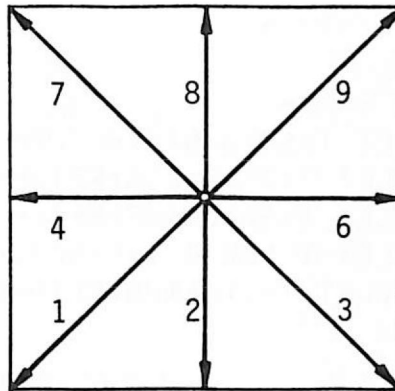
U, V	:	Aktueller Stand des Zeichenstiftes
U(I), V(I)	:	Felder der Dimension I zur Speicherung der Koordinaten mehrerer Punkte
UU(I), VV(I)	:	Felder der Dimension I zur Speicherung von Koordinaten nach einer Koordinatentransformation (vergl. Abs. 3.1.4)
T	:	Schrittweite beim Ziehen mit GLCURSOR
N	:	Zahl der Punkte eines Linienzuges
AL	:	Winkel α bei der Koordinatentransformation
W	:	Winkelschrittweite beim Zeichnen von Kreisen und Ellipsen (vergl. Abs. 3.1.10 ... 3.1.12)
W1, W2	:	Anfangs- und Endwinkel beim Zeichnen von Kreis- und Ellipsenbogen (vergl. Abs. 3.1.10 ... 3.1.12)
RA, RB	:	Radien bzw. Halbmesser bei Kreisen und Ellipsen
M	:	Nummer der Grafik-Aufgabe

3.1 GRAFIK-GRUNDAUFGABEN

3.1.1 Verschieben des Zeichenstiftes

Die erste Aufgabe ist das komfortable Verschieben des Zeichenstiftes ohne eine Linie zu zeichnen. Dazu wird mit den Zifferntasten des Rechners die Zugrichtung festgelegt. Die BASIC - INKEY\$ - Funktion ermöglicht die elegante Eingabe der Zugrichtung.

Die Ziffern



geben durch ihre Anordnung die Zugrichtung an.

```
100 T=1:U=0:V=0:H=200:B=200
110 "INFO1"A=ASC INKEY$ : IF A=0GOTO 110
120 IF A>47AND A<58GOSUB "ZUG"
130 GOTO "INFO1"
```

```
2400 REM -----
2410 "ZUG"
2420 REM -----
2430 IF A=48LET U=0:V=0
2440 IF A=49LET U=U-T:V=V-T
2450 IF A=50LET V=V-T
2460 IF A=51LET U=U+T:V=V-T
2470 IF A=52LET U=U-T
2480 IF A=53GOTO "TEMPO"
```

Do not sale !


```
2490 IF A=54LET U=U+T
2500 IF A=55LET U=U-T:U=U+T
2510 IF A=56LET U=U+T
2520 IF A=57LET U=U+T:U=U+T
2530 IF U<0LET U=0
2540 IF U>BLET U=B
2550 IF U<0LET U=0
2560 IF U>HLET U=H
2570 IF D$=""GLCURSOR (U,U)
2580 IF D$<>"LPRINT "M";U;"",";U
2590 RETURN
2600 REM -----
2610 "TEMPO"
2620 REM -----
2630 IF T=1LET T=10:G=&14:RETURN
2640 IF T=10LET T=25:G=&2A:RETURN
2650 IF T=25LET T=50:G=&3E:RETURN
2660 IF T=50LET T=100:G=&7F:RETURN
2670 IF T=100LET T=1:G=&08:RETURN
```

Bei diesem BASIC-Programm erkennt man die Sonderrolle der Ziffern 0 und 5: 0 führt zum Koordinatenursprung zurück, ($U = 0$, $V = 0$), 5 ändert die "Geschwindigkeit" der Stiftbewegung.

Es wäre sehr lästig, immer in Einer-Schritten über die Bildfläche ziehen zu müssen. Durch wiederholtes Betätigen der 5 - Taste kann man die Geschwindigkeit nach den in "TEMPO" angegebenen T-Werten wählen.

B und H sind die maximale Breite und Höhe der Zeichung. G ist ein Displayzeichencode zur Darstellung der Geschwindigkeit mit GPRINT (vergl. Grafik-Editor GRADOR).

3.1.2 Zeichnen von Linienzügen und Vielecken

Unter einer Linie soll hier die geradlinige Verbindung zwischen zwei Punkten verstanden werden. Ein Linienzug ist die Aneinander-

Do not sale !

reihung mehrerer Linien.

Linien werden mit dem LINE-Befehl bzw. mit LPRINT "D" gezeichnet. Mit dem folgenden BASIC-Programm kann man beliebige Linien oder Linienzüge zeichnen, wenn in den Feldern U(...) und V(...) die Koordinaten der einzelnen Linienpunkte gespeichert sind. N gibt die Zahl der Punkte des Linienzuges an. Der Anfangspunkt des Linienzuges wird in U(0) und V(0) gespeichert.

```
100 DIM U(3),V(3)
110 N=2
120 U(0)=10:V(0)=-20
130 U(1)=200:V(1)=20
140 GOSUB "LINIE"
150 END

5360 REM -----
5370 "LINIE"
5380 REM -----
5390 IF D$<>"GOTO "LI1"
5400 GLCURSOR (U(0),V(0))
5410 FOR I=0TO N-1:LINE -(U(I),V(I)),L:NEXT I
5420 IF M=2LINE -(U(0),V(0)),L
5430 RETURN
5440 "LI1"LPRINT "M";U(0);", ";V(0)
5450 FOR I=1TO N-1:LPRINT "D";U(I);", ";V(I):NEXT I
5460 IF M=2LPRINT "D";U(0);", ";V(0)
5470 RETURN
```

Als Ergebnis des kleinen Hauptprogramms sehen Sie eine einfache Linie:



Das Unterprogramm "LINIE" dient also zum Zeichnen von Linien und Linienzügen.

Do not sale !

Wenn man beim obigen Beispiel das Hauptprogramm durch ein anderes

```
100 DIM U(3),U(3)
110 N=4
120 U(0)=10:U(0)=-20
130 U(1)=200:U(1)=20
132 U(2)=200:U(2)=-50
134 U(3)=20:U(3)=-40
140 GOSUB "LINIE"
150 END
```

ersetzt, dann wird ein Linienzug gezeichnet:



Wenn ein Linienzug wieder zum Anfangspunkt zurückkehrt, wird die Zeichenfigur geschlossen: Es entsteht ein Vieleck. Um auch Vielecke mit "LINIE" zeichnen zu können, ist im Unterprogramm eine Variable M eingeführt. Wenn M = 2 ist, dann wird der Linienzug zum Vieleck geschlossen.

```
100 DIM U(3),U(3)
110 N=4
115 M=2
120 U(0)=10:U(0)=-20
130 U(1)=200:U(1)=20
132 U(2)=200:U(2)=-50
134 U(3)=20:U(3)=-40
140 GOSUB "LINIE"
150 END
```



Do not sale !

Mancher Leser wird sicherlich fragen, was das ganze soll. Man könnte ja das Beispiel von oben viel einfacher eingeben als

LINE (10,-20) - (200,20) - (200,-50) - (20,-40) - (10,-20)

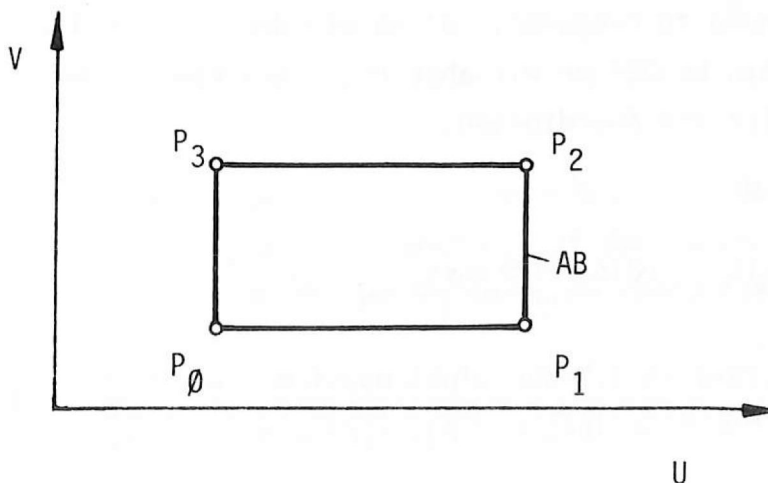
oder

LINE (U(0),V(0)) - (U(1),V(1)) - (U(2),V(2)) - (U(3),V(3)) -
(U(4),V(4)) - (U(0),V(0))

Das ist bestimmt richtig. Problematisch wird es aber, wenn in einem Programm viele Linienzüge oder Vielecke mit unterschiedlicher Zahl von Punkten gezeichnet werden sollen. Mit den eben gezeigten Zeilen kann man immer nur ein Viereck zeichnen, kein Dreieck oder Fünfeck. Man müßte dann für jedes n-Eck oder für jeden Linienzug eine eigene Programmzeile vorsehen, was auf Dauer sicher nicht von Vorteil ist.

3.1.3 Zeichnen von Rechtecken: Teil 1

Mit dem Unterprogramm "LINIE" lassen sich natürlich Rechtecke zeichnen, wenn man die Koordinaten der vier Rechteckpunkte kennt. Rechtecke möchte man aber oft nur durch die Grundseite und die Rechteckbreite bestimmen:

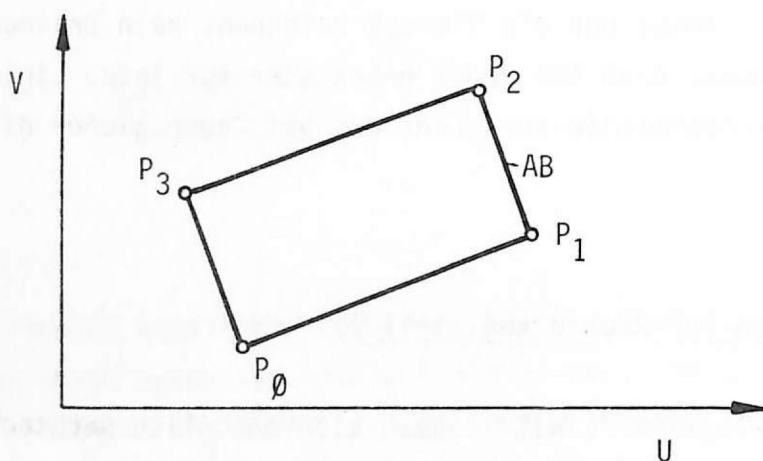


Do not sale !

Gegeben sind also P_0 und P_1 durch $U(0)$, $V(0)$ und $U(1)$, $V(1)$ und die Breite AB der Rechteckseite. Wenn das Rechteck so wie im Bild in der U,V -Ebene liegt, dann sind

$$\begin{aligned} U(2) &= U(1) & V(2) &= V(1) + AB \\ U(3) &= U(0) & V(3) &= V(2) \end{aligned}$$

einfach bestimmbar. Das Rechteck kann vom Unterprogramm "LINIE" gezeichnet werden. Kompliziert wird es aber, wenn das Rechteck "schräg" im Koordinatensystem liegt:

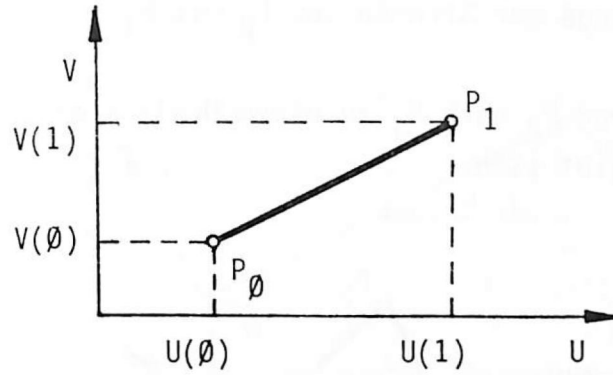


Die Punkte P_2 und P_3 sind jetzt zwar immer noch berechenbar, die Gleichung dazu wird aber sehr kompliziert. Um uns die Aufstellung der Bestimmungsgleichung zu ersparen, wollen wir dem Rechner diese Arbeit übertragen. Dazu benötigen wir aber erst noch Kenntnisse über die Transformation von Koordinaten.

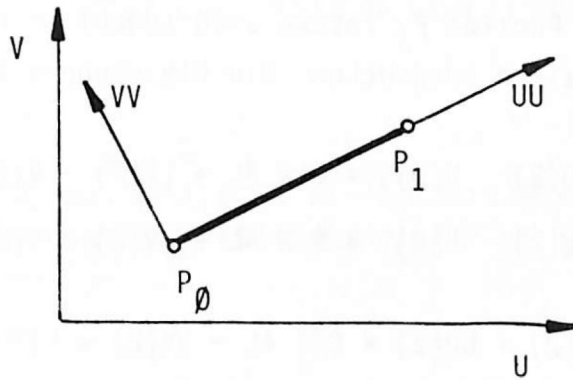
3.1.4 Zweidimensionale Koordinatentransformation

Wir betrachten eine Linie im U,V -Koordinatensystem. Die Punkte P_0 und P_1 werden durch ihre Koordinaten $U(0)$, $V(0)$ und $U(1)$, $V(1)$

Do not sale !



bestimmt. Man kann natürlich auch bei gleicher Lage ein anderes Koordinatensystem mit den Koordinaten UU und VV wählen:



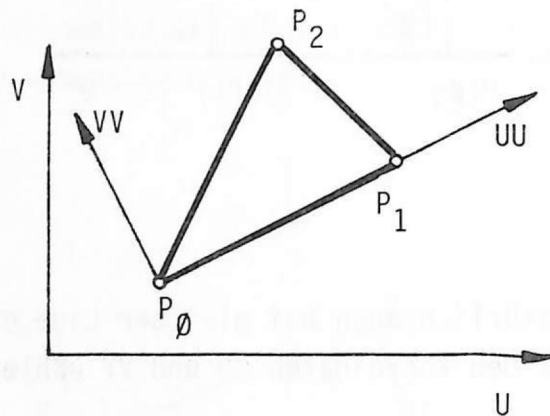
Das neue Koordinatensystem wird so gelegt, daß der Koordinatennullpunkt im Punkt P_0 und der Punkt P_1 auf der UU -Achse liegt. Die Koordinaten der Punkte P_0 und P_1 im neuen Koordinatensystem sind jetzt sehr einfach:

Do not sale !

$$\begin{array}{ll} UU(\emptyset) = \emptyset & VV(\emptyset) = \emptyset \\ UU(1) = s & VV(1) = \emptyset \end{array}$$

mit s = Länge der Strecke von P_{\emptyset} bis P_1 .

Wenn wir die Linie von P_{\emptyset} nach P_1 zu einem Dreieck mit dem Punkt P_2 erweitern, so ergibt sich



Die Koordinaten des Punktes P_2 lassen sich sowohl im neuen Koordinatensystem wie im alten ausdrücken. Die Gleichungen lauten:

$$UU(2) = (U(2) - U(\emptyset)) * \cos AL + (V(2) - V(\emptyset)) * \sin AL$$

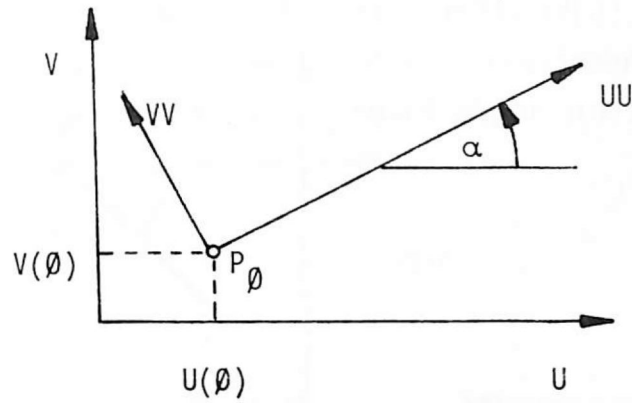
$$VV(2) = -(U(2) - U(\emptyset)) * \sin AL + (V(2) - V(\emptyset)) * \cos AL$$

oder:
$$U(2) = U(\emptyset) + UU(2) * \cos AL - VV(2) * \sin AL$$

$$V(2) = V(\emptyset) + UU(2) * \sin AL + VV(2) * \cos AL$$

Mit diesen Gleichungen lassen sich entweder die UU, VV -Koordinaten berechnen, wenn die U, V -Koordinaten bekannt sind, oder umgekehrt. Die Transformation $U \rightarrow UU$ erfolgt mit dem Unterprogramm "U->UU", die Rücktransformation mit "UU->U". Man muß zur Transformation nur die Lage der Koordinatensystem-Nullpunkte zueinander ($U(\emptyset)$ und $V(\emptyset)$) und den Drehwinkel $\alpha = \text{ALPHA} = \text{AL}$ zwischen der U - und der UU -Achse kennen.

Do not sale !



```
1100 REM -----  
1110 "U->UU"  
1120 REM -----  
1130 UU(J)=(U(J)-U(0))*COS AL+(V(J)-V(0))*SIN AL  
1140 VV(J)=-(U(J)-U(0))*SIN AL+(V(J)-V(0))*COS AL  
1150 RETURN  
1160 REM -----  
1170 "UU->U"  
1180 REM -----  
1190 U(J)=U(0)+UU(J)*COS AL-VV(J)*SIN AL  
1200 V(J)=V(0)+UU(J)*SIN AL+VV(J)*COS AL  
1210 RETURN
```

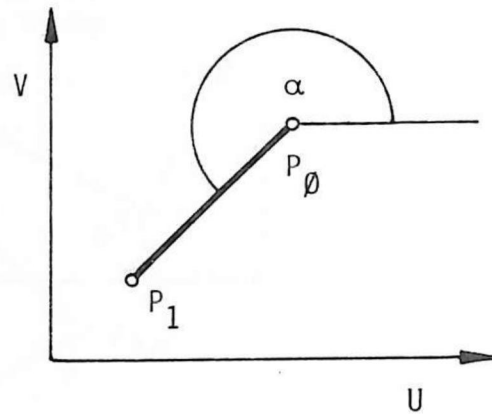
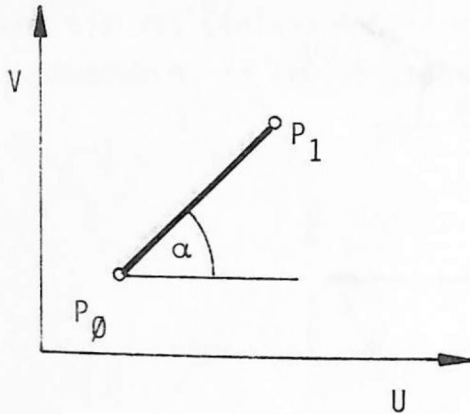
α ergibt sich aus der Forderung, daß der Punkt P_0 Ursprung des UU, VV -Koordinatensystems sein soll und P_1 auf der UU -Achse liegen soll. α ist der Steigungswinkel der Strecke von P_0 nach P_1 im U, V -Koordinatensystem:

$$\tan \alpha = \frac{V(1) - V(0)}{U(1) - U(0)}$$

Falls $U(1) = U(0)$ ist, so wird $\alpha = \pm 90^\circ$.

Do not sale !

α wird so gewählt, daß $-90^{\circ} < \alpha < 90^{\circ}$, wenn $U(1) > U(\emptyset)$.
Wenn $U(1) < U(\emptyset)$, so wird $90^{\circ} < \alpha < 270^{\circ}$.



Diese Gleichungen sind in den BASIC-Unterprogrammen "TRANS" und "STEIGUNG" des Grafik-Editors enthalten.

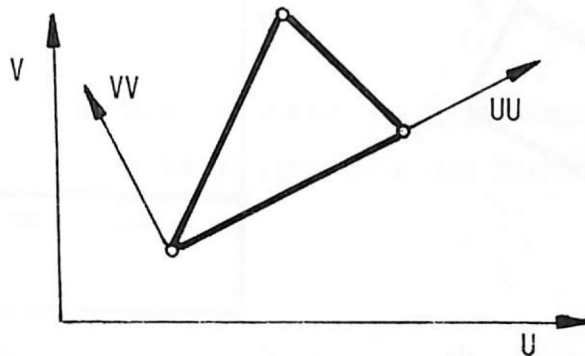
```
1010 REM -----
1020 "TRANS"
1030 REM -----
1040 UU(0)=0:VV(0)=0:UU(J)=0
1050 I=0:GOSUB "STEIGUNG"
1060 IF AL=90LET UU(J)=V(J)-V(0):RETURN
1070 IF AL=-90LET UU(J)=V(0)-V(J):RETURN
1080 UU(J)=(U(J)-U(0))/COS AL
1090 RETURN

1220 REM -----
1230 "STEIGUNG"
1240 REM -----
1250 IF U(I)=U(J)AND V(J)>V(I)LET AL=90:RETURN
1260 IF U(I)=U(J)AND V(J)<V(I)LET AL=-90:RETURN
1270 AL=ATN ((V(J)-V(I))/(U(J)-U(I)))
1280 IF U(I)>U(J)LET AL=AL+180
1290 RETURN
```

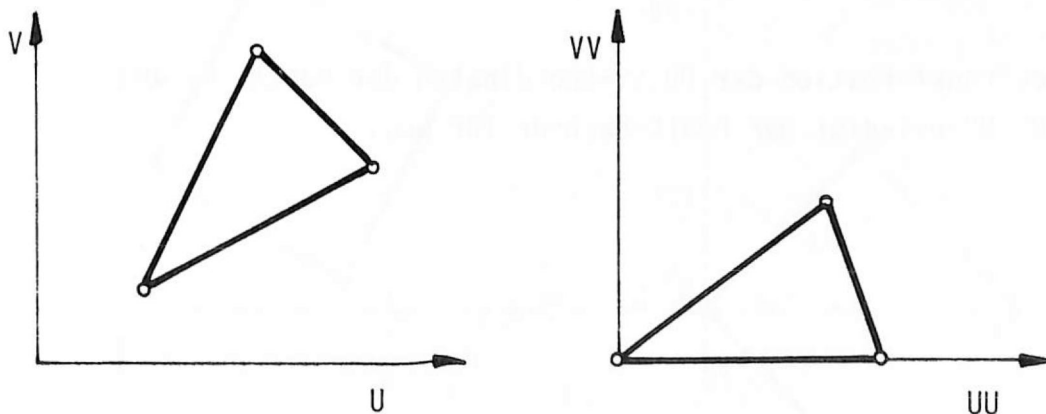
Do not sale !

Mit dem Unterprogramm "TRANS" sind wir also in der Lage, ein neues Koordinatensystem so zu legen, daß der Punkt P_0 einer Strecke Koordinatennullpunkt und der Punkt P_j dieser Strecke Punkt der neuen UU-Achse wird. Nachdem "TRANS" einmal aufgerufen wurde, kann man mit dem Unterprogramm "U->UU" die UU(J), VV(J) - Koordinaten bei Vorgabe der Koordinaten U(J) und V(J) bestimmen. Mit dem Unterprogramm "UU->U" kann man die U,V-Koordinaten eines Punktes P_j bei gegebenen UU,VV-Koordinaten berechnen.

Man kann übrigens vereinfachend das Bild



auch in zwei Teilbilder zerlegen, wobei beide Koordinatensysteme nebeneinander gezeichnet werden:

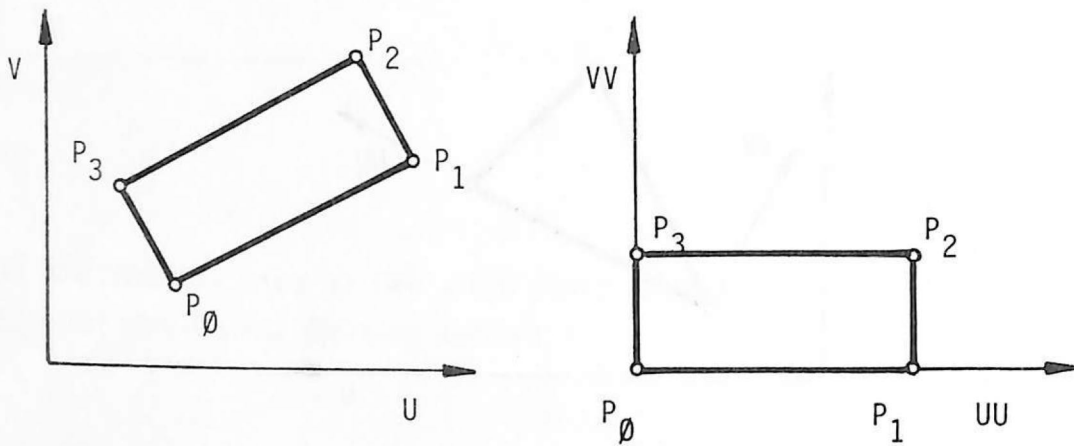


Do not sale !

Diese Darstellungsweise wird bei den weiteren Anwendungen der Transformationsformeln bevorzugt werden.

3.1.5 Zeichnen von Rechtecken: Teil 2 (Konstruktion "REC" bei GRADOR)

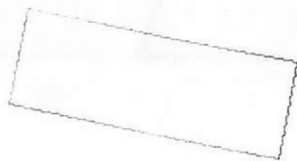
Doch zurück zu unserem Rechteck. Wir transformieren das Rechteck in das UU,VV-Koordinatensystem:



Die Koordinaten der Punkte P_2 und P_3 ergeben sich ganz einfach zu

$$\begin{array}{ll} UU(2) = UU(1) & VV(2) = AB \\ UU(3) = \emptyset & VV(3) = VV(2) \end{array}$$

Die Rücktransformation der UU,VV-Koordinaten der Punkte P_2 und P_3 mit "UU->U" erledigt der BASIC-Rechner für uns.



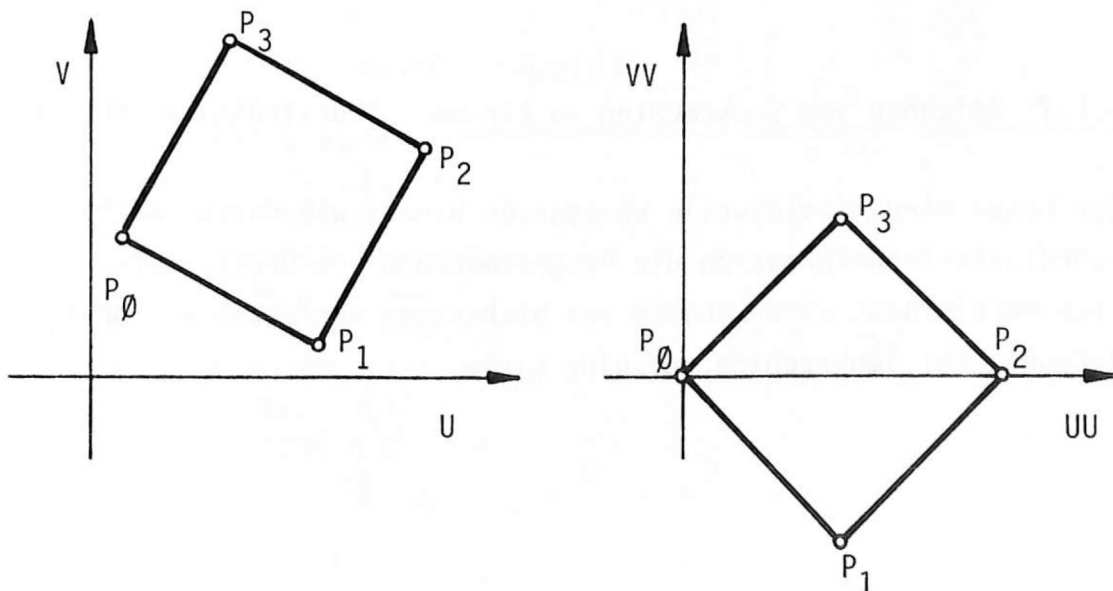
Do not sale !

```
120 DIM U(3),V(3),UU(3),VV(3)
110 N=4:M=2
120 U(0)=30:U(0)=-50
130 U(1)=200:U(1)=-80
140 AB=60
150 J=1:GOSUB "TRANS"
160 UU(2)=UU(1):VV(2)=AB
170 UU(3)=0:VV(3)=AB
180 J=2:GOSUB "UU->U"
190 J=3:GOSUB "UU->U"
200 GOSUB "LINIE"
210 END
```

Der Leser sieht also, wie ihm durch geeignete Programmierung die relativ komplizierte Berechnung der fehlenden Rechteckkoordinaten vom Rechner abgenommen wird.

3.1.6 Zeichnen von Quadraten (Konstruktion "QUA" bei GRADOR)

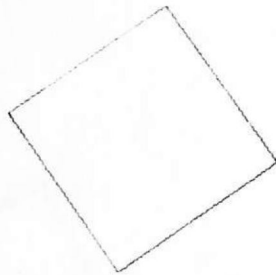
Ähnlich wie beim Rechteckzeichnen gehen wir beim Zeichnen von Quadraten vor:



Do not sale !

Gegeben sind die Punkte P_0 und P_2 eines Quadrats. Durch geeignete Transformation ergeben sich

$$\begin{array}{ll} UU(0) = 0 & VV(0) = 0 \\ UU(1) = UU(2) / 2 & VV(1) = - UU(1) \\ UU(2) \text{ aus Transf.} & VV(2) = 0 \\ UU(3) = UU(1) & VV(3) = - VV(1) \end{array}$$

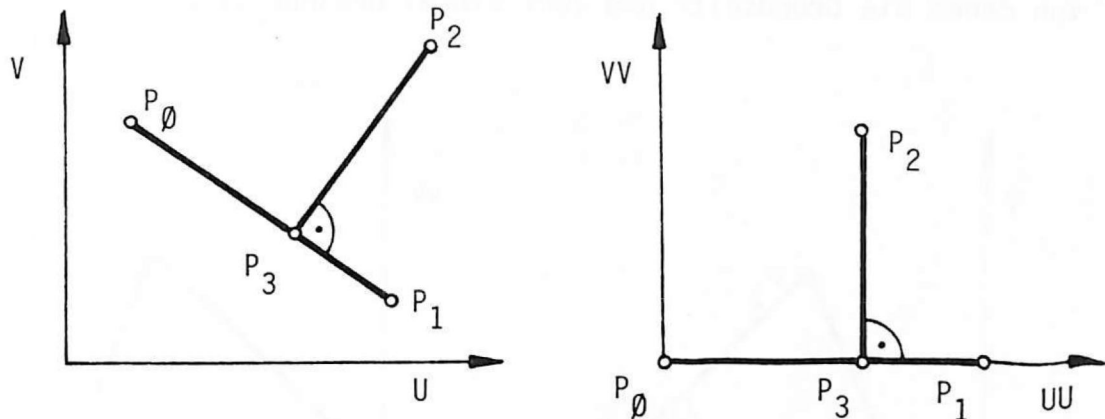


```
100 DIM U(3),V(3),UU(3),VV(3)
110 N=4:M=2
120 U(0)=30:V(0)=-50
130 U(2)=200:V(2)=-80
150 J=2:GOSUB "TRANS"
160 UU(1)=UU(2)/2:VV(1)=-UU(1)
170 UU(3)=UU(1):VV(3)=-VV(1)
180 J=1:GOSUB "UU->U"
190 J=3:GOSUB "UU->U"
200 GOSUB "LINIE"
210 END
```

3.1.7 Zeichnen von Senkrechten zu Linien (Konstruktion "SEN" bei GRADOR)

Der Leser wird mittlerweile verstanden haben, wie durch geschickte Koordinatentransformation die Programmierung von Grafik-Aufgaben wesentlich vereinfacht wird. Analog zum bisherigen verfahren wir beim Zeichnen der Senkrechten auf eine Linie:

Do not sale !



Gegeben sind P_0 , P_1 und P_2 . Gesucht ist der Punkt P_3 zum Zeichnen der Senkrechten von P_2 nach P_3 .

Die Koordinaten ergeben sich zu

$$UU(3) = UU(2) \quad VV(3) = \emptyset$$

Also:

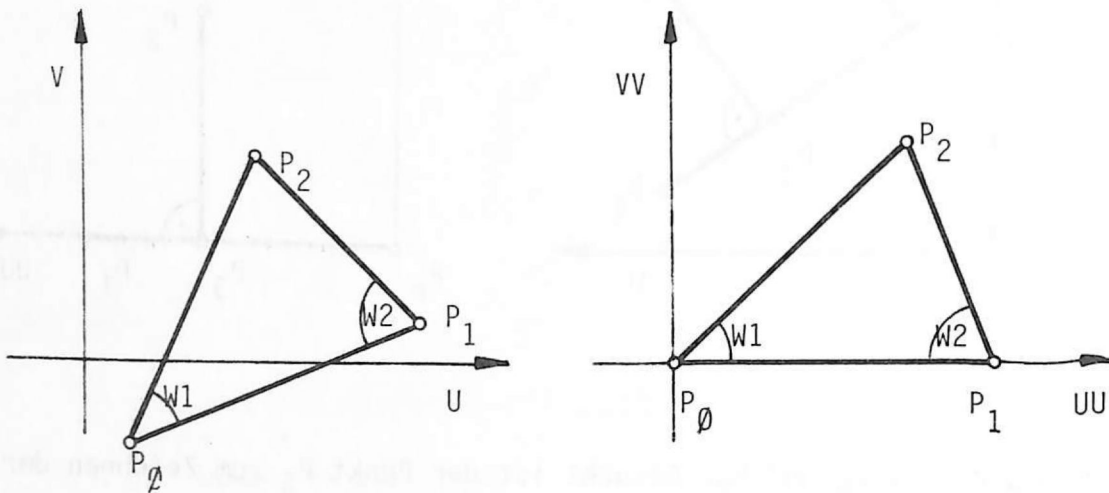
```
100 DIM U(3),V(3),UU(3),VV(3)
110 N=2:M=1
120 U(0)=30:V(0)=-50
130 U(1)=200:V(1)=-80
135 GOSUB "LINIE"
140 U(2)=130:V(2)=-10
150 J=1:GOSUB "TRANS"
160 J=2:GOSUB "U->UU"
170 UU(0)=UU(2):VV(0)=0
180 J=0:GOSUB "UU->U"
190 U(1)=U(2):V(1)=V(2)
200 GOSUB "LINIE"
210 END
```



Do not sale !

3.1.8 Zeichnen von Dreiecken (Konstruktion "DRE" bei GRADOR)

Mit dem weiter oben besprochenen Unterprogramm "LINIE" kann man beliebige Dreiecke zeichnen, wenn man die Koordinaten aller Dreieckspunkte kennt. Teilweise wird man aber Dreiecke zeichnen wollen, von denen die Grundseite und zwei Winkel bekannt sind.



Um das UP "LINIE" anwenden zu können, müssen die Koordinaten des dritten Punktes berechnet werden. Nach einer geeigneten Koordinatentransformation kann man eine Gleichung zur Berechnung von P_2 verwenden, die in entsprechenden Mathematikwerken (z.B. BRONSTEIN / SEMENDJAJEW) zu finden ist:

$$W3 = 180 - W1 - W2$$

$$UU(2) = UU(1) * \sin W2 / \sin W3 * \cos W1$$

$$VV(2) = UU(1) * \sin W2 / \sin W3 * \sin W1$$

In BASIC ergibt sich folgendes Programm:



Do not sale !

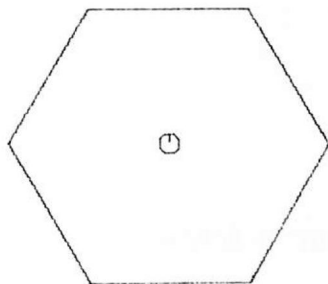
```
100 DIM U(3),U(3),UU(3),UU(3)
110 N=3:M=2
120 U(0)=30:U(0)=-50
130 U(1)=200:U(1)=-80
140 W1=45:W2=30
150 W3=180-W1-W2
160 J=1:GOSUB "TRANS"
170 UU(2)=UU(1)*SIN W2/SIN W3*COS W1
180 UU(2)=UU(1)*SIN W2/SIN W3*SIN W1
190 J=2:GOSUB "UU->U"
200 GOSUB "LINIE"
210 END
```

3.1.9 Zeichnen von Kreisen und Kreisbogen: Teil 1

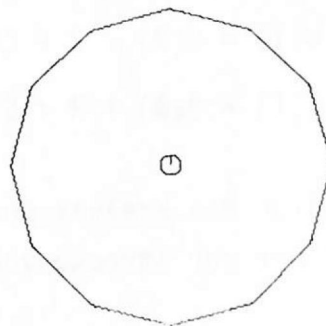
(Konstruktionen "K-1" und "B-1" bei GRADOR)

Bislang haben wir nur Linien und Vielecke gezeichnet. Wir wollen uns nun mit dem Zeichnen von Kreisen beschäftigen. (Besitzer des CE-516P können den Abschnitt überspringen, da das Zeichnen von Kreisen in diesem Plotter bereits implementiert ist.)

Einen Kreis zeichnet man mit einem Plotter, indem man den Kreis durch ein Vieleck annähert. Die folgenden Bilder zeigen, daß ein 6- oder 12-Eck sicherlich keine gute Näherung ist:

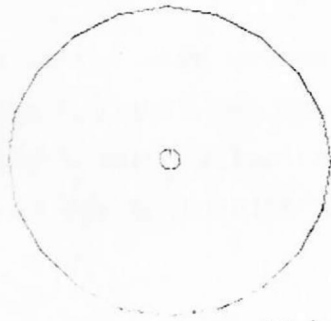


6-Eck

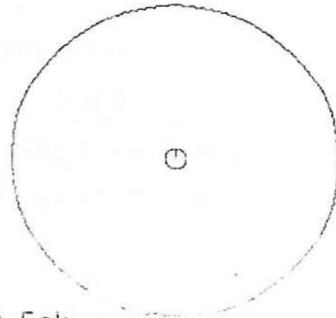


12-Eck

Do not sale !



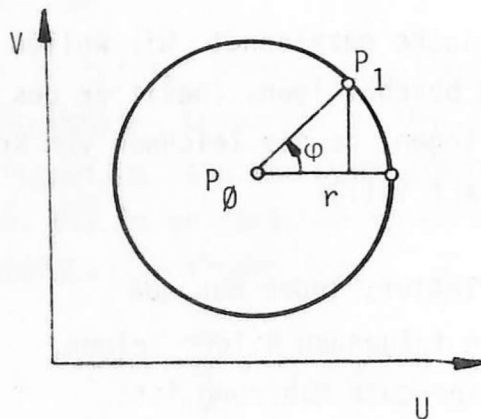
20-Eck



40-Eck

Je nach Auflösungsgrad des Plotters und nach Größe des Kreises genügen aber 20 ... 40 Punkte auf der Kreislinie zum Zeichnen des Kreises.

Punkte auf der Kreislinie kann man berechnen als



$$U(1) = U(0) + R * \cos \varphi$$

$$V(1) = V(0) + R * \sin \varphi$$

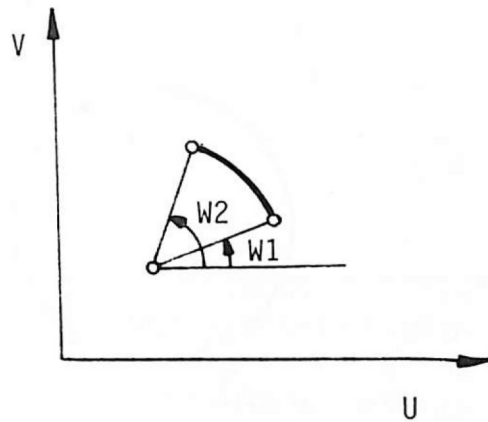
mit R = Radius des Kreises und

φ = Winkel der Strecke von P_0 nach P_1 zur U-Achse.

Zum Zeichnen eines Vollkreises bestimmt man also z.B. 36 Punkte, indem φ in 10er-Schritten von $\varphi = 0^0$ bis $\varphi = 360^0$ läuft.

Do not sale !

Wenn man nur einen Kreisbogen von W1 bis W2 zeichnen will, so läßt man ϕ eben von W1 bis W2 laufen.



Das Unterprogramm "KREIS" zum Zeichnen von Kreisen und Kreisbogen lautet:

```
6560 REM -----
6570 "KREIS"
6580 REM -----
6590 IF W1>W2LET W2=W2+360:GOTO 6590
6600 PH=W1
6610 X=U(0)+RA*COS PH
6620 Y=U(0)+RA*SIN PH
6630 IF D$=""GLCURSOR (X,Y)
6640 IF D$<>""LPRINT "M";X;",";Y
6650 "NEXT PH"
6660 PH=PH+W
6670 IF PH>W2LET PH=W2
6680 X=U(0)+RA*COS PH
6690 Y=U(0)+RA*SIN PH
6700 IF D$=""LINE -(X,Y),L
6710 IF D$<>""LPRINT "D";X;",";Y
6720 IF PH<W2GOTO "NEXT PH"
6730 RETURN
```

RA = Radius

PH = Winkel ϕ

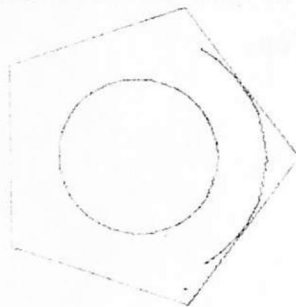
W1 = Anfangswinkel

W2 = Endwinkel

W = Winkelschrittweite

Do not sale !

Beispiel:



```
100 DIM U(0),V(0)
110 U(0)=100:V(0)=-100
120 RA=50
130 W1=0:W2=360:W=10
140 GOSUB "KREIS"
150 RA=80
160 W1=-60:W2=60
170 GOSUB "KREIS"
180 RA=100
190 W1=0:W2=360:W=72
200 GOSUB "KREIS"
210 END
```

3.1.10 Zeichnen von Kreisen: Teil 2

Im letzten Abschnitt wurden Kreise gezeichnet, bei denen der Mittelpunkt und der Radius gegeben sind. Es sind aber auch andere Kreisvorgaben denkbar:

Gegeben sind zwei gegenüberliegende Punkte des Kreises

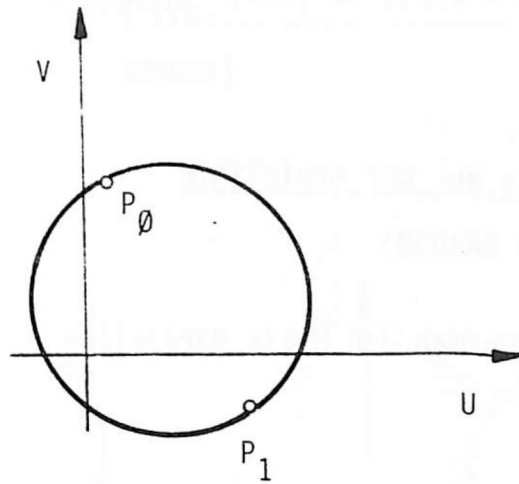
(Konstruktion "K-2" bei GRADOR)

Der Radius (vergl. Bild auf der nächsten Seite) berechnet sich zu

$$RA = 0.5 * \sqrt{(U(1) - U(0))^2 + (V(1) - V(0))^2}$$

Der Mittelpunkt ergibt sich zu

Do not sale !



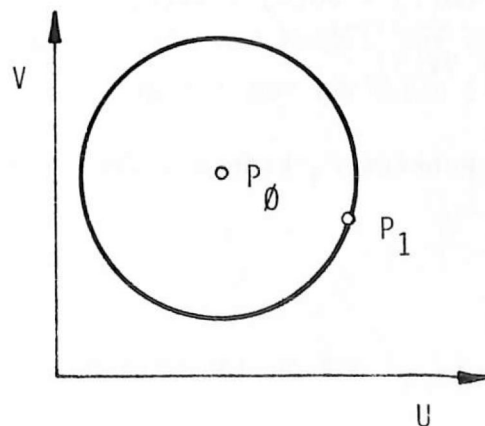
$$U(0) = 0.5 * (U(0) + U(1))$$

$$V(0) = 0.5 * (V(0) + V(1))$$

Danach kann man mit dem UP "KREIS" den Kreis zeichnen.

Gegeben sind der Mittelpunkt und ein Punkt auf der Kreislinie

(Konstruktion "K-3" bei GRADOR)



Do not sale !

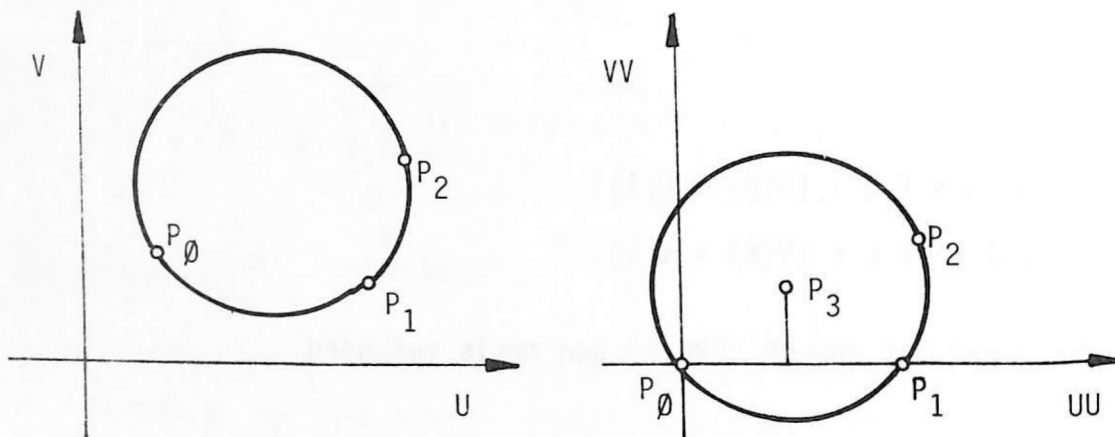
Der Radius berechnet sich zu

$$RA = \sqrt{(U(1) - U(\emptyset))^2 + (V(1) - V(\emptyset))^2}$$

Gegeben sind drei Punkte auf der Kreislinie

(Konstruktion "K-4" bei GRADOR)

Durch Transformation kann man den Kreis darstellen als



Der Mittelpunkt liegt bei

$$UU(3) = UU(1) / 2$$

$$VV(3) = (UU(2)^2 - UU(1) * UU(2) + VV(2)^2) / (2 * VV(2))$$

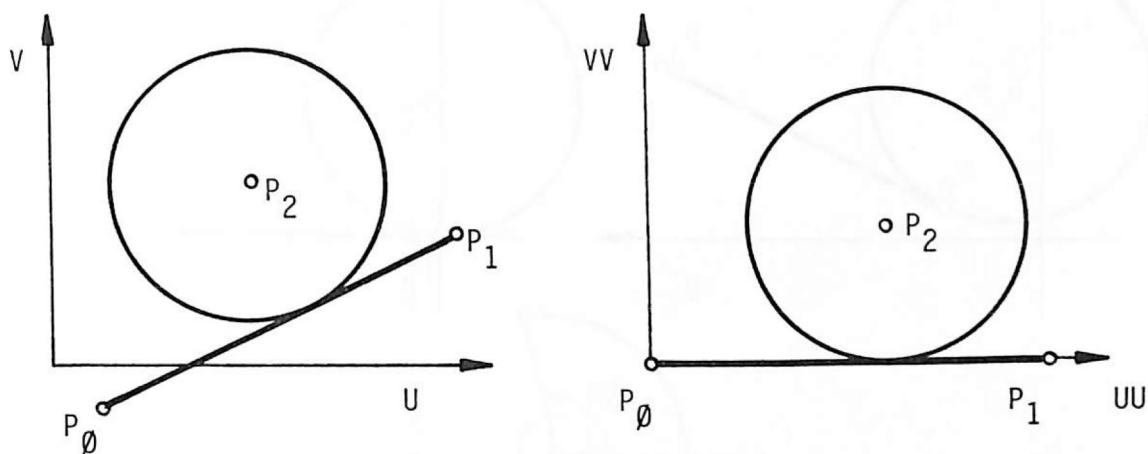
$$RA = \sqrt{UU(3)^2 + VV(3)^2}$$

Nach Rücktransformation des Punktes P₃ kann man den Kreis zeichnen.

Do not sale !

Gegeben ist eine Linie durch 2 Punkte und der Mittelpunkt des die Linie berührenden Kreises-----

(Konstruktion "K-5" bei GRADOR)



Nach einer Transformation ergibt sich für den Radius

$$RA = | VV(2) |$$

Gegeben ist eine Linie durch zwei Punkte und der Radius des die Linie im Punkt P_0 berührenden Kreises-----

(Konstruktion "K-6" bei GRADOR)

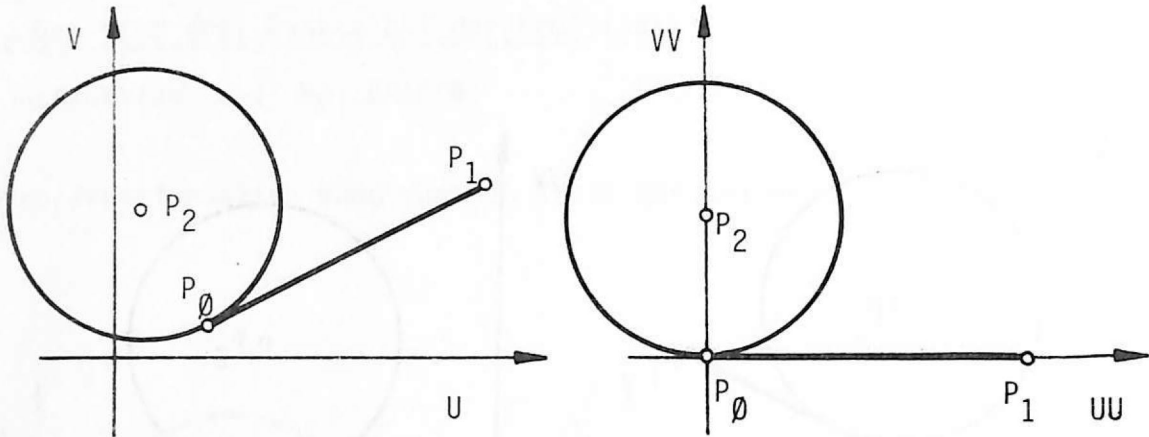
Den Kreismittelpunkt kann man nach einer entsprechenden Transformation (vergleiche Bild auf der nächsten Seite) als

$$UU(2) = \emptyset$$

$$VV(2) = RA$$

finden. Nach Rücktransformation von P_2 kann man den Kreis zeichnen.

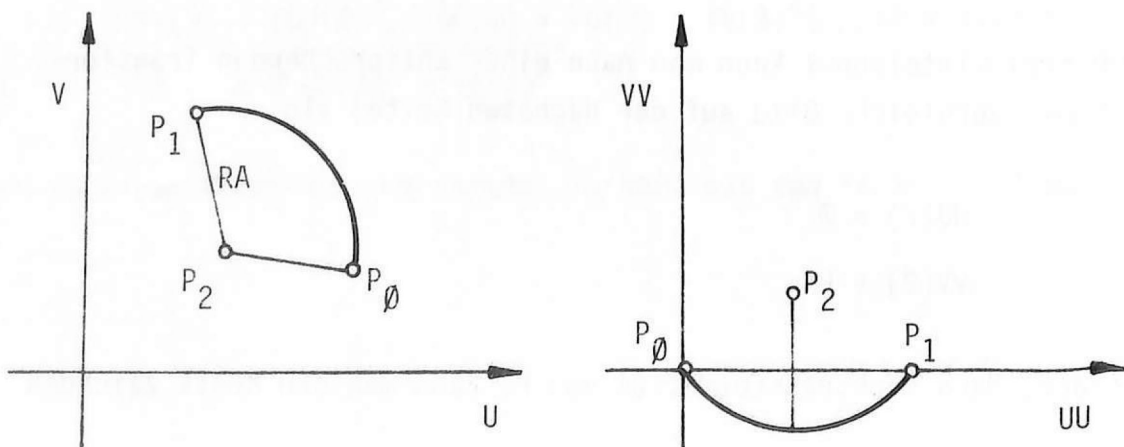
Do not sale !



3.1.11 Zeichnen von Kreisbogen: Teil 2

Im Abschnitt 3.1.9 wurde ein Kreisbogen gezeichnet, bei dem Mittelpunkt, Radius und zwei Winkel gegeben sind. Zwei weitere Konstruktionen sollen besprochen werden:

Gegeben sind Anfangs- und Endpunkt des Kreisbogens und der Radius
(Konstruktion "B-2" bei GRADOR)

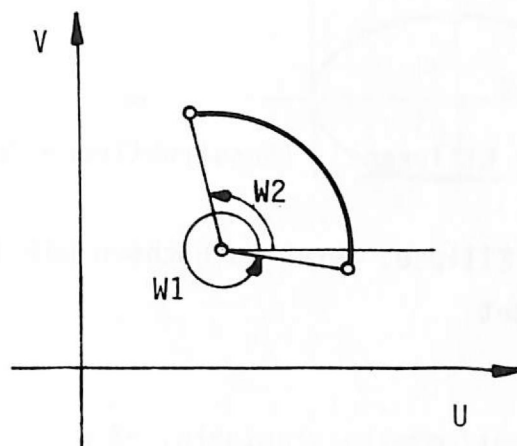


Do not sale !

Nach entsprechender Transformation ergibt sich für P_2 :

$$UU(2) = UU(1) / 2$$
$$VV(2) = \sqrt{RA^2 - UU(2)^2}$$

Nach der Rücktransformation ist der Mittelpunkt bekannt. Den Anfangs- und Endwinkel des Kreisbogens bestimmen wir mit dem Unterprogramm "STEIGUNG".



Gegeben sind der Mittelpunkt und der Anfangs- und Endpunkt
des Kreisbogens

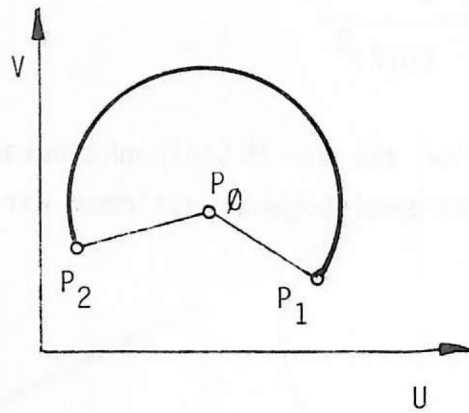
(Konstruktion "B-3" bei GRAODR)

Den Radius (vergleiche Bild auf der nächsten Seite) berechnet man
zu

$$RA = \sqrt{(U(1) - U(\emptyset))^2 + (V(1) - V(\emptyset))^2}$$

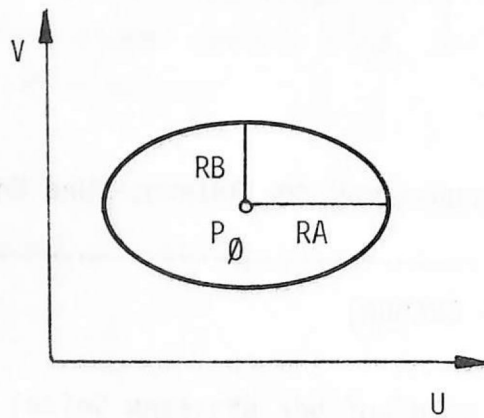
Danach kann man wie zuvor den Anfangs- und Endwinkel des Kreis-
bogens bestimmen.

Do not sale !



3.1.12 Zeichnen von Ellipsen (Konstruktionen "E-1" und "E-2")

Die Gleichung einer Ellipse, deren Halbachsen parallel zur U- bzw. V-Achse liegen, lautet



$$U(1) = U(\emptyset) + RA * \text{COS } \varphi$$

$$V(1) = V(\emptyset) + RB * \text{SIN } \varphi$$

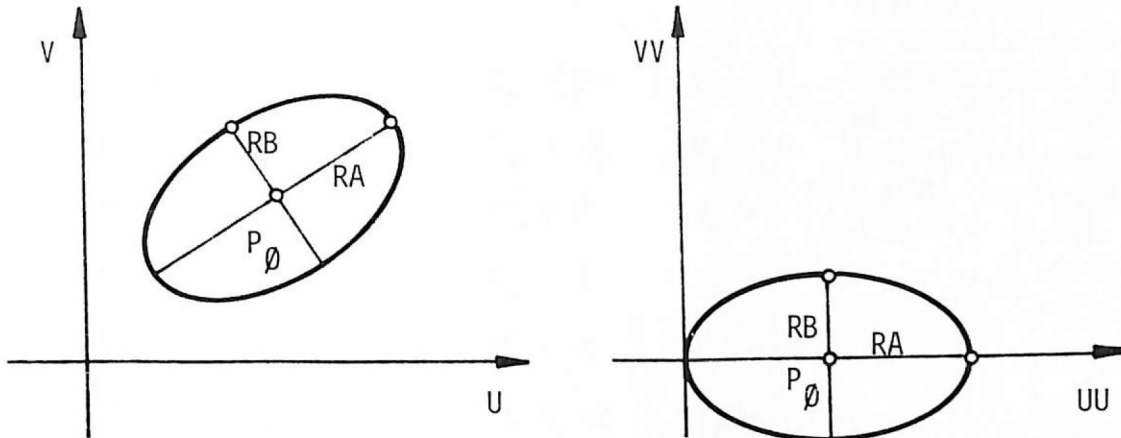
Man kann also im Programm zum Kreiszeichnen (Abs. 3.1.9) die Zeile 662 \emptyset durch

$$Y = V(\emptyset) + RB * \text{SIN } PH$$

Do not sale !

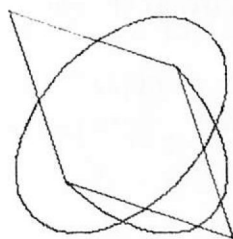
ersetzen, und schon zeichnet das Programm eine Ellipse.

Problematisch wird es, wenn die Ellipse schräg in der U,V-Ebene liegt:



Hier vermeidet wieder die Koordinatentransformation das Erstellen komplizierter Formeln. In der UU,VV-Ebene wird die gewünschte Ellipse Punkt für Punkt berechnet. Jeder Punkt wird rücktransformiert und zum Zeichnen verwendet. (Konstruktion "E-1" bei GRADOR)
Durch Vorgabe verschiedener Anfangs- und Endwinkel kann man auch Ellipsenbogen zeichnen. (Konstruktion "E-2" bei GRADOR)

Achtung: Das Programm verlangt die Vorgabe der Winkel an der transformierten Ellipse!



Do not sale !

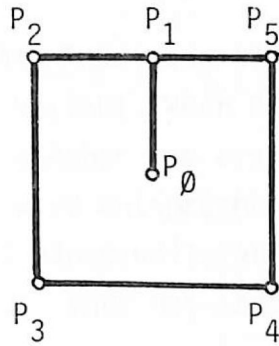
```
100 DIM U(1),V(1),UU(1),VV(1)
110 U(0)=100:V(0)=-100
115 AL=-45
120 RA=50:RB=80
130 W1=0:W2=360:W=10
140 GOSUB "ELLIPSE"
150 RA=80:RB=50
160 W1=-90:W2=90
170 GOSUB "ELLIPSE"
180 RA=100:RB=50
190 W1=0:W2=360:W=90
200 GOSUB "ELLIPSE"
210 END
```

3.1.13 Zeichnen von Sonderzeichen

Nach diesem recht umfangreichen Ausflug in die Geometrie wollen wir uns wieder mit etwas einfacherem beschäftigen: Mit der Markierung von Punkten. Zuweilen möchte man bestimmte Punkte in der Zeichenebene hervorheben, indem man einen Kreis oder ein anderes Symbol an diese Stellen setzt. Folgende Sonderzeichen sollen verwendet werden:

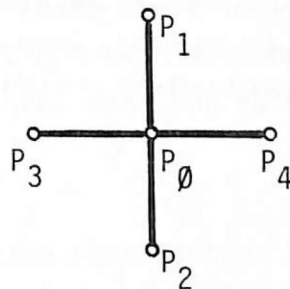


Diese Zeichen lassen sich einfach mit dem RLINE-Kommando erzeugen. Bei RLINE gibt man bekanntlich den Zielpunkt einer zu zeichnenden Linie durch die Angabe relativer Koordinaten vor.



Von P_0 nach P_1	:	$U_r = \emptyset$	$V_r = 2$
Von P_1 nach P_2	:	$U_r = -2$	$V_r = \emptyset$
Von P_2 nach P_3	:	$U_r = \emptyset$	$V_r = -4$
Von P_3 nach P_4	:	$U_r = 4$	$V_r = \emptyset$
Von P_4 nach P_5	:	$U_r = \emptyset$	$V_r = 4$
Von P_5 nach P_1	:	$U_r = -2$	$V_r = \emptyset$

Oder:

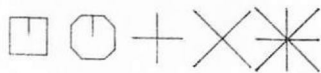


Bei diesem Beispiel darf man die Linie von P_2 nach P_3 nicht zeichnen, die Linie von P_0 nach P_1 braucht man nicht zu zeichnen. Das muß bei der Vorgabe der Koordinaten berücksichtigt werden.

Die relativen Koordinaten der Linien kann man gut in DATA-Zeilen speichern. Das folgende Programm ermöglicht das Zeichnen der oben gezeigten Zeichen. Die entsprechenden relativen Koordinaten stehen in den DATA-Zeilen "SON-1" ... "SON-5".

Do not sale !

```
3900 REM -----
3910 "SOZEI"
3920 REM -----
3930 RESTORE "SON-"+CHR$(N+32)
3940 "SO-1"
3950 READ I: IF I=999RETURN
3960 IF I>0GOTO "SO-2"
3970 I=-I
3980 READ J,K
3990 IF D$=""RLINE (J*S,K*S)-(0,0),9
4000 IF D$<>""LPRINT "R";J*S;",";K*S
4010 "SO-2"
4020 FOR Q=1TO I
4030 READ J,K
4040 IF D$=""RLINE -(J*S,K*S),L
4050 IF D$<>""LPRINT "J";J*S;",";K*S
4060 NEXT Q
4070 GOTO "SO-1"
4080 REM -----
4090 "SON-1"DATA 6,0,2,-2,0,0,-4,4,0,0,4,-2,0,999
4100 "SON-2"DATA 10,0,2,-1,0,-1,-1,0,-2,1,-1,2,0,1,1,0,2,-1
,1,-1,0,999
4110 "SON-3"DATA -1,0,2,0,-4,-1,-2,2,4,0,999
4120 "SON-4"DATA -1,-2,-2,4,4,-1,-4,0,4,-4,999
4130 "SON-5"DATA -1,-2,2,4,-4,-1,-4,0,4,4,-1,-2,0,0,-4,-1,-
2,2,4,0,999
```



```
100 U=-20:V=-50
110 S=5:N=16
120 FOR II=0TO 4
130 U=U+41:S=S+1
140 IF D$=""GLCURSOR (U,V)
150 IF D$<>""LPRINT "M";U;",";V
160 N=N+1
170 GOSUB "SOZEI"
180 NEXT II
190 END
```

Do not sale !

Die erste Zahl in einer DATA-Anweisung gibt die Zahl der Punkte an, aus der ein Linienzug besteht. Wenn die erste Zahl negativ ist, dann fährt der Stift gehoben zum ersten Punkt. Wenn ein Linienzug gezeichnet ist, wird entsprechend der nächste begonnen, bis die erste Zahl eines weiteren Linienzuges = 999 ist. Das ist für das Programm das Ende des Sonderzeichens.

3.2 DER GRAFIK - EDITOR GRADOR

Alle im Abschnitt 3.1 vorgestellten Grafik-Grundkonstruktionen sind im sogenannten Grafik-Editor GRADOR enthalten. Zweck von GRADOR ist es, das Erstellen von Bildern aus diesen Konstruktionen zu vereinfachen. Über eine entsprechende Menüsteuerung kann man z.B. angeben, welche Grundkonstruktion an welcher Stelle gezeichnet werden soll. Darüber hinaus ermöglicht es der Grafik-Editor, alle Bildelemente zu speichern. Man kann dann beliebig oft ein gespeichertes Bild zeichnen lassen, man kann es ändern oder auf Kassette abspeichern.

Der auf den folgenden Seiten beschriebene Grafik-Editor ist sehr komfortabel, aber auch sehr lang. Er ist für den PC-1500 (A) geschrieben. Man benötigt einen BASIC-Speicher von mindestens 15 kByte. Leser, die ihren PC-1500 (A) nicht genügend weit ausgebaut haben, können den Grafik-Editor nur mit Tricks anwenden. Zunächst sollte man alle REM-Zeilen löschen. Der Grafik-Editor belegt dann 11517 Bytes. Weiterhin besteht die Möglichkeit, auf verschiedene Grundkonstruktionen zu verzichten. Dazu sind einige Programmzeilen speziell markiert. Beim Weglassen der Zeilen mit dem Zeichen Δ entfällt das Zeichnen des Hilfsrasters, beim Weglassen der Zeilen mit \circ entfallen die Sonderzeichen. Die Ellipsenkonstruktionen sind mit \square markiert. Der Verzicht auf alle eben genannten Konstruktionen reduziert die GRADOR-Größe auf 9836 Bytes. Weiterhin können Anwender, die nur den CE-150 oder nur den CE-515P / 516P verwenden

Do not sale !

wollen, alle Zeilen löschen, die für den fremden Plotter benötigt werden.

Leser, die nicht den PC-1500 (A) besitzen, können das Programm-
listing dazu verwenden, für ihren Rechner ein ähnliches Programm
zu erstellen.

Bevor einige programmtechnische Details erklärt werden, soll ein
Musterlauf gezeigt werden. GRADOR wird mit RUN "A" gestartet.

DISPLAY	EINGABE
Externer Drucker ?	J oder N, je nach verwendetem Drucker. CE-150 : N CE-515P / 516P : J
Daten von Band?	J : Bild wird von Band gelesen. Es muß zuvor gespeichert worden sein. Der Rekorder muß an der richtigen Stelle stehen und ein- geschaltet sein. N : Erzeugen eines neuen Bildes
UMAX = ?	Eingabe der maximalen Bildbreite z.B. : 200 beim CE-150 400 beim CE-515P / 516P Achtung: CE-150 : UMAX \leq 220 CE-515P / 516P : UMAX \leq 960
VMAX = ?	Dito für die Bildhöhe
Rahmen ?	Frage, ob zur besseren Orientierung ein Rahmen um das Bild gezeichnet werden soll.
C 0 R 0 S 2 L 0 0 0	1. Menü : Gibt die aktuellen Werte der Farbe C (COLOR), der Schreibrichtung R (ROTATE), der Schriftgröße S (CSIZE),


Do not sale !

der Linienart L (LINE) und die Lage des Stiftes in U- und V-Koordinaten an. Man kann durch Betätigen der entsprechenden Reservetasten C, R, S und L ändern. Der Zeichenstift wird mit den Zifferntasten 0 ... 9 (vergl. Abs. 3.1.1) bewegt. Die Punkte zwischen den U- und V-Koordinaten geben die Bewegungsgeschwindigkeit an. Mit der RCL-Taste kann man das Winkelinkrement beim Zeichnen von Kreisen und Ellipsen bestimmen.

↓ : Bewegen des Zeichenblattes, um das bisher gezeichnete Bild besser betrachten zu können


↑ : Blatt wieder zurück

Mit der DEF-Taste kann man das Speichern der zuletzt gezeichneten Grundkonstruktion veranlassen.

Mit  gelangt man ins Menü 2

L-N V-N DRE REC QUA SEN

Auswahl einer entsprechenden Grundkonstruktion

Weiter mit  zum Menü 3 (Zum Menü 1 gelangt man immer mit der RCL-Taste)


K-1 K-2 K-3 K-4 K-5 K-6

Auswahl der entsprechenden Kreiskonstruktionen

Weiter mit  zum Menü 4

B-1 B-2 B-3 E-1 E-2

Auswahl von Kreisbogen- und Ellipsenkonstruktionen

Weiter mit  zum Menü 5

Do not sale !

RAS TEX SON

RAS : Zeichnen eines Rasters zu Orientierungszwecken

TEX : Drucken eines Textes

SON : Zeichnen eines Sonderzeichens

Weiter mit


TES ZEI SAV END

TES : Zeichnen des gespeicherten Bildes mit Abfrage nach jeder Teilkonstruktion, ob sie gelöscht werden soll


ZEI : Zeichnen des ganzen Bildes ohne Unterbrechung

SAV : Speichern des Bildes auf Kassette. Rekorder muß eingeschaltet sein.

END : Ende des GRADOR-Laufs

Mit  gelangt man wieder zum Menü 1

Sobald man sich für eine Konstruktion entschieden hat (Drücken der entsprechenden Reservetaste), erfragt das Programm alle benötigten Daten. Als erstes wird ausgegeben, wieviele Punkte schon eingegeben wurden und wieviele insgesamt benötigt werden. Zur Punkteingabe fährt man mit dem Stift an die entsprechende Stelle und drückt die ENTER-Taste: Die aktuelle Stiftposition wird als Punkt gespeichert. Die Reihenfolge der Punkte bei der Punkteingabe entnehmen Sie bitte den Beschreibungen der Grundkonstruktionen.

Beginnen wir also mit dem Zeichnen: Gehen Sie zunächst zum Menü 1 mit  oder der RCL-Taste. Wir wollen ein Rechteck zeichnen. Drücken der entsprechenden Reservetaste im Menü 2: Auf dem Display erscheint $\emptyset \dots 2$. Das gibt an, das noch kein Punkt eingegeben wurde und daß insgesamt 2 Punkte eingegeben werden müssen. Stift auf $(5\emptyset, 5\emptyset)$, ENTER, Stift auf $(25\emptyset, 75)$, ENTER. Die Frage nach dem Abstand beantworten Sie mit $5\emptyset$. Der Plotter zeichnet das

Do not sale !

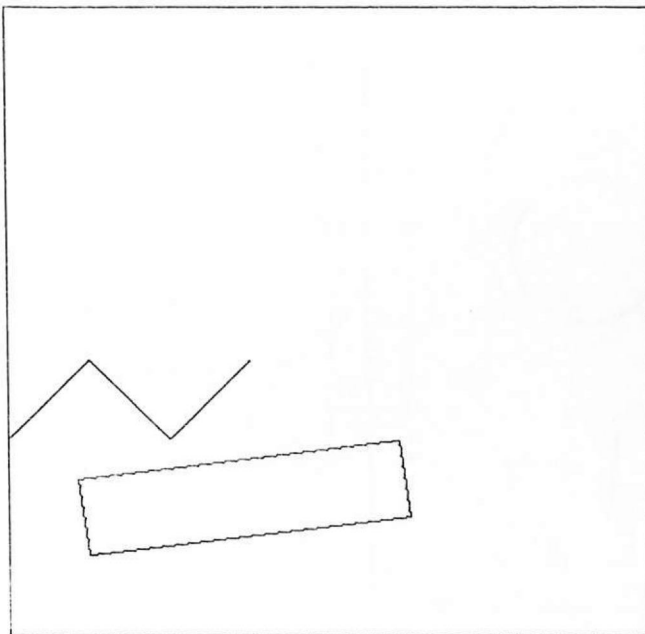
Rechteck. Wenn die Konstruktion gespeichert werden soll, betätigt man die DEF-Taste, sobald das Menü 1 wieder erscheint.

Zeichnen einer Linie aus 4 Punkten: L-N. Beantworten Sie die Frage nach der Zahl der Punkte mit 4. Dann fahren Sie an die entsprechenden Punkte und geben sie mit ENTER ein.

Z.B.: U = 0 V = 125
 U = 50 V = 175
 U = 100 V = 125
 U = 150 V = 175

Der Linienzug wird gezeichnet. Speichern mit DEF.

Unser Bild sieht bis jetzt wie folgt aus:



Zeichnen eines Kreises aus 3 Punkten: K-4. Das Display verlangt die Eingabe der 3 Punkte. Z.B.:

Do not sale !

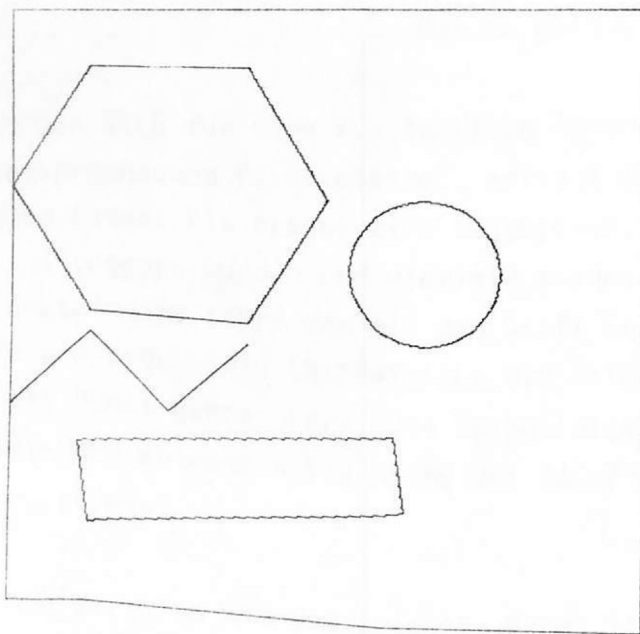
U = 285	V = 200
U = 300	V = 200
U = 275	V = 275

Wenn man das Winkelinkrement beim Kreiszeichnen ändern will, so betätigt man vom Menü 1 aus die RCL-Taste. Wir ändern die Voreinstellung von 10° auf 60° .

Weiteres Beispiel für einen Kreis: Mittelpunkt und Radius sind gegeben: K-1.

U = 100	V = 275	RA = 100
---------	---------	----------

Man sieht: Es wird ein regelmäßiges Sechseck gezeichnet, da daß Winkelinkrement auf $60^{\circ} = 360^{\circ} / 6$ festgelegt wurde.



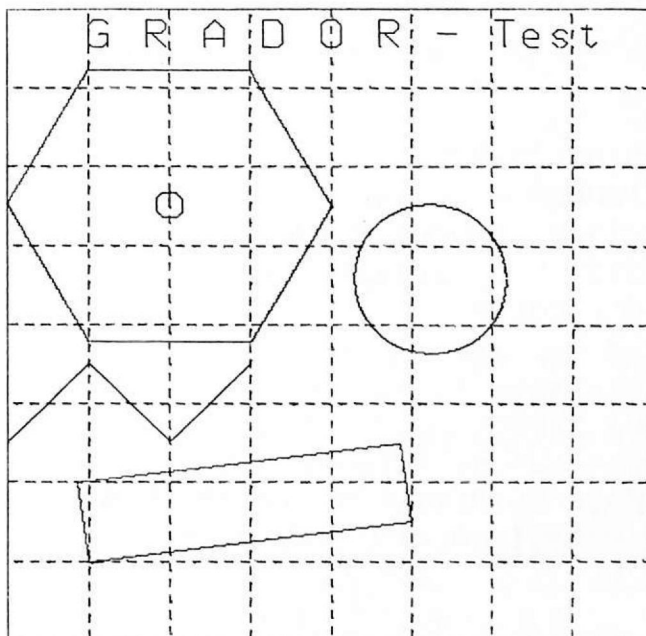
Zur besseren Orientierung auf der Zeichenebene kann man ein Raster zeichnen lassen. Zuvor ändern wir den Linientyp auf 3 im Menü 1. Die Rasterkonstruktion wird mit RAS im Menü 5 aufgerufen. Der Rechner fragt nach dem Rasterabstand: 50. Das Raster, das nun gezeichnet

Do not sale !

wird, kann allerdings nicht gespeichert werden.

Eingabe eines Sonderzeichens beim Kreismittelpunkt: Stift an die Stelle $U = 100$, $V = 275$ und Eingabe von SON im Menü 5. Die fünf möglichen Sonderzeichen erscheinen im Display. Man wählt eines (z.B. das zweite) durch Drücken der entsprechenden Reservetaste. Je nach Größe von S (CSIZE) wird das Sonderzeichen unterschiedlich groß gezeichnet.

Text als Bildüberschrift: Zuerst Kontrolle von S: $S = 3$. Stift an die entsprechende Stelle bewegen: $U = 50$, $V = 375$. Drücken der TEX-Taste im Menü 5. Die Texteingabe wird verlangt: GRADOR - Test.



Wenn Sie das Bild anschauen wollen, so betätigen Sie die TES-Taste im letzten Menü. Das Bild wird gezeichnet, falls Sie nicht DEF irgendwann vergessen haben. Nach dem Zeichnen jeder Teilkonstruktion erscheint die Frage LOE WEI END

LOE : Die eben gezeichnete Teilkonstruktion wird aus dem Speicher gelöscht

Do not sale !

WEI : Weiterzeichnen

END : Zurück zum Menü 1, ohne den Rest des Bildes zeichnen zu lassen.

Soweit die Beschreibung eines Musterlaufes. Sie können GRADOR verlassen, indem Sie im letzten Menü die END-Reservetaste betätigen. Am besten spielen Sie etwas mit GRADOR, um alle Möglichkeiten des Grafik-Editors kennenzulernen. Sie haben natürlich die Möglichkeit, einige Grundkonstruktionen aus dem Programm zu entfernen, wenn Sie keinen Bedarf daran haben, andere neue können Sie analog zu den bisherigen einprogrammieren.

```
1000 REM GRAFIK - EDITOR GRADOR
1010 REM -----
1020 "TRANS"
1030 REM -----
1040 UU(0)=0:VV(0)=0:UU(J)=0
1050 I=0:GOSUB "STEIGUNG"
1060 IF AL=90LET UU(J)=V(J)-V(0):RETURN
1070 IF AL=-90LET UU(J)=V(0)-V(J):RETURN
1080 UU(J)=(U(J)-U(0))/COS AL
1090 RETURN
1100 REM -----
1110 "U->UU"
1120 REM -----
1130 UU(J)=(U(J)-U(0))*COS AL+(V(J)-V(0))*SIN AL
1140 VV(J)=- (U(J)-U(0))*SIN AL+(V(J)-V(0))*COS AL
1150 RETURN
1160 REM -----
1170 "UU->U"
1180 REM -----
1190 U(J)=U(0)+UU(J)*COS AL-VV(J)*SIN AL
1200 V(J)=V(0)+UU(J)*SIN AL+VV(J)*COS AL
1210 RETURN
1220 REM -----
1230 "STEIGUNG"
1240 REM -----
1250 IF U(I)=U(J)AND V(J)>V(I)LET AL=90:RETURN
1260 IF U(I)=U(J)AND V(J)<V(I)LET AL=-90:RETURN
1270 AL=ATN ((V(J)-V(I))/(U(J)-U(I)))
1280 IF U(I)>U(J)LET AL=AL+180
1290 RETURN
```

Do not sale !

```
1300 REM -----
1310 REM Start von GRADOR
1320 REM -----
1330 "A"
1340 REM -----
1350 D1=20:D2=20:D3=20:D4=20
1360 I1=0:I2=0:I3=0:I4=0
1370 DIM SP$(D1,D2)*1,SP(D3,D4)
1380 UU=10
1390 DIM U(UU),V(UU),UU(UU),VV(UU),TE$(0)*80
1400 REM -----
1410 WAIT 0:INPUT "EXTERNER DRUCKER ? (J/N) ";D$:IF D$<>"J"
LET D$=""
1420 REM -----
1430 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PO:CONSOLE
0,0
1440 INPUT "DATEN VON BAND ? (J/N) ";B$:IF B$="J"INPUT #I1,
I2,I3,I4,SP$(*),SP(*)
1450 DEGREE :CLS
1460 IF D$=""GRAPH
1470 IF D$<>" "LPRINT CHR$(27)+"b"
1480 REM -----
1490 REM Voreinstellungen
1500 REM -----
1510 C=0:S=2:L=0:T=1:G=&08:U=0:V=0:E=0:R=0:W=10
1520 IF D$=""COLOR C:CSIZE S:ROTATE R
1530 IF D$<>" "LPRINT CHR$(27)+CHR$(830+C):LPRINT CHR$(27
)+"?"+CHR$(860+S):LPRINT "Q";R:LPRINT "L";L
1540 INPUT "UMAX = ? ";B:INPUT "UMAX = ? ";H
1550 IF D$=""GLCURSOR (0,-H):SORGN
1560 IF D$<>" "LPRINT "M0,";-H:LPRINT "I"
1570 INPUT "RAHMEN ? (J/N) ";R$
1580 IF R$="J"AND D$=""LINE (0,0)-(B,H),0,B
1590 IF R$="J"AND D$<>" "LPRINT "D0,0,";B;"",0,";B;"",";H;"",0,
";H;"",0,0"
1600 GOTO "MENUE0"
1610 REM -----
1620 "PKTEIN"
1630 REM -----
1640 IF N>(UU+1)WAIT 100:PRINT "N ZU GROSS":WAIT 0:GOTO "ME
NUE0"
1650 I=N:E=1:GOTO "MENUE0"
```

Do not sale !

```
1660 REM -----
1670 "MENUE00"
1680 REM -----
1690 E=0
1700 IF D$=""GLCURSOR (U,U)
1710 IF D$<>"LPRINT "M";U;"",";U
1720 REM -----
1730 "MENUE0"
1740 REM -----
1750 DI=0:CLS
1760 REM -----
1770 "MENUE1"
1780 REM -----
1790 CURSOR 16:PRINT USING "#####";U;U;
1800 CURSOR 1
1810 IF E=0AND DI=0PRINT USING "##";"C";C;" R";R;:PRINT USI
NG "#####";S;:CURSOR 12:PRINT L;:CURSOR 9:PRINT "S";
1820 IF E=0AND DI=0CURSOR 13:PRINT "L";
1830 IF E=0AND DI=1PRINT USING "#####.#";"W";W;" "
1840 IF E=1PRINT USING "#####";E$;N-I;N;
1850 GCURSOR 1:GPRINT &7F;:GPRINT &7F:IF DI=0GCURSOR 26:GPR
INT &7F;:GPRINT &7F
1860 IF E=0GCURSOR 50:GPRINT &7F;:GPRINT &7F:GCURSOR 74:GPR
INT &7F;:GPRINT &7F
1870 GCURSOR 98:GPRINT &7F;:GPRINT &7F:GCURSOR 128:GPRINT G
;:GPRINT G
1880 USING
1890 REM -----
1900 "INFO1"A=ASC INKEY$ :IF A=0GOTO 1900
1910 REM -----
1920 IF A>47AND A<58GOSUB "ZUG":GOTO "MENUE1"
1930 IF E=1GOTO "INFO11"
1940 IF A=9GOTO "MENUE2"
1950 IF A=25AND DI=0LET DI=1:GOTO "MENUE1"
1960 IF A=25AND DI=1LET DI=0:GOTO "MENUE1"
1970 IF DI=1AND A=17INPUT "WINKELINKREMENT ? ";W:GOTO "MENU
E0"
1980 IF DI=0ON A-16GOTO "COLOR","ROTATE","CSIZE","LINE"
1990 IF A=27GOTO "SPEICHER"
2000 GOTO "INFO12"
2010 "INFO11"
2020 IF A<>13GOTO "INFO12"
2030 U(N-I)=U:U(N-I)=U:I=I-1:IF I=0GOTO E$
2040 GOTO "MENUE1"
```

Do not sale !

```
2050 "INFO12
2060 IF A=10AND D$=""GLCURSOR (U,U-200):GOTO "MENUE1
2070 IF A=10AND D$<>"LPRINT "M";U;",";U+200:GOTO "MENUE1
2080 IF A=11AND D$=""GLCURSOR (U,U):GOTO "MENUE1
2090 IF A=11AND D$<>"LPRINT "M";U;",";U:GOTO "MENUE1
2100 GOTO "INFO1"
2110 REM -----
2120 "COLOR"
2130 REM -----
2140 INPUT "COLOR (0 ... 3) ? ";C:IF C<0OR C>3GOTO 2140
2150 IF D$=""COLOR C
2160 IF D$<>"LPRINT CHR$ (27)+CHR$ (&30+C)
2170 GOTO "MENUE0"
2180 REM -----
2190 "CSIZE"
2200 REM -----
2210 IF D$=""INPUT "CSIZE (1 ... 9) ? ";S:IF S<1OR S>9GOTO
2210
2220 IF D$<>"INPUT "CSIZE (1 ... 15) ? ";S:IF S<1OR S>15GO
TO 2220
2230 IF D$=""CSIZE S
2240 IF D$<>"LPRINT CHR$ (27)+"?" +CHR$ (&60+S)
2250 GOTO "MENUE0"
2260 REM -----
2270 "ROTATE"
2280 REM -----
2290 INPUT "ROTATE (0 ... 3) ? ";R:IF R<0OR R>3GOTO 2290
2300 IF D$=""ROTATE R
2310 IF D$<>"LPRINT "Q";R
2320 GOTO "MENUE0"
2330 REM -----
2340 "LINE"
2350 REM -----
2360 IF D$=""INPUT "LINE (0 ... 8) ? ";L:IF L<0OR L>8GOTO 2
360
2370 IF D$<>"INPUT "LINE (0 ... 15) ? ";L:IF L<0OR L>15GOT
O 2370
2380 IF D$<>"LPRINT "L";L
2390 GOTO "MENUE0"
2400 REM -----
2410 "ZUG"
2420 REM -----
2430 IF A=48LET U=0:V=0
2440 IF A=49LET U=U-T:V=V-T
```

Do not sale !


```
2450 IF A=50LET U=U-T
2460 IF A=51LET U=U+T:V=V-T
2470 IF A=52LET U=U-T
2480 IF A=53GOTO "TEMPO"
2490 IF A=54LET U=U+T
2500 IF A=55LET U=U-T:V=V+T
2510 IF A=56LET U=U+T
2520 IF A=57LET U=U+T:V=V+T
2530 IF U<0LET U=0
2540 IF U>BLET U=B
2550 IF U<0LET U=0
2560 IF U>HLET U=H
2570 IF D$=""GLCURSOR (U,V)
2580 IF D$<>"LPRINT "M";U;",";U
2590 RETURN
2600 REM -----
2610 "TEMPO"
2620 REM -----
2630 IF T=1LET T=10:G=&14:RETURN
2640 IF T=10LET T=25:G=&2A:RETURN
2650 IF T=25LET T=50:G=&3E:RETURN
2660 IF T=50LET T=100:G=&7F:RETURN
2670 IF T=100LET T=1:G=&08:RETURN
2680 REM -----
2690 "SPEICHER"
2700 REM -----
2710 PAUSE "SPEICHERN"
2720 SP$(I1,I2)=CHR$ M:GOSUB "SP$"
2730 SP$(I1,I2)=CHR$ C:GOSUB "SP$"
2740 IF M=10R M=20R M=8LET SP$(I1,I2)=CHR$ N:GOSUB "SP$"
2750 IF M<>7LET SP$(I1,I2)=CHR$ L:GOSUB "SP$"
2760 IF M=7LET F=LEN TE$(0):SP$(I1,I2)=CHR$ F:GOSUB "SP$"
2770 IF M=70R M=8LET SP$(I1,I2)=CHR$ S:GOSUB "SP$"
2780 IF M=7LET SP$(I1,I2)=CHR$ R:GOSUB "SP$"
2790 IF M<>1AND M<>2LET N=1
2800 IF M=70R M=8LET U(0)=U:U(0)=U
2810 FOR I=0TO N-1:Q=U(I):GOSUB "SP-P":Q=U(I):GOSUB "SP-P":
NEXT I
2820 IF M=30R M=40R M=50R M=6LET SP(I3,I4)=RA:GOSUB "SP":SP
(I3,I4)=W:GOSUB "SP"
2830 IF M=50R M=6LET SP(I3,I4)=RB:GOSUB "SP":SP(I3,I4)=AL:G
OSUB "SP"
2840 IF M=40R M=6LET SP(I3,I4)=W1:GOSUB "SP":SP(I3,I4)=W2:G
OSUB "SP"
2850 IF M=7FOR I=1TO F:SP$(I1,I2)=MID$(TE$(0),I,1):GOSUB "
SP$":NEXT I
2860 GOTO "MENUE0"
```

Do not sale !

```
2870 REM -----
2880 "SP$" I2=I2+1: IF I2>D2LET I2=0: I1=I1+1
2890 RETURN
2900 REM -----
2910 "SP" I4=I4+1: IF I4>D4LET I4=0: I3=I3+1
2920 RETURN
2930 REM -----
2940 "SP-P"
2950 V0=0: IF Q<0LET V0=1
2960 P1=INT ABS (Q/256)
2970 P2=ABS Q-256*P1
2980 IF V0=1LET P1=P1+2^7
2990 SP$(I1, I2)=CHR$ P1: GOSUB "SP$"
3000 SP$(I1, I2)=CHR$ P2: GOSUB "SP$"
3010 RETURN
3020 REM -----
3030 "MENUE2"
3040 REM -----
3050 PRINT " L-N U-N DRE REC QUA SEN"
3060 "INFO2" A=ASC INKEY$ : IF A=0GOTO 3060
3070 IF A=9GOTO "MENUE3"
3080 IF A=25GOTO "MENUE0"
3090 IF A<17OR A>22GOTO "INFO2"
3100 IF A=17LET E$="L-N"
3110 IF A=18LET E$="U-N"
3120 IF A=17OR A=18INPUT "Zahl der Punkte ? "; N: IF N<2GOTO
3120
3130 IF A=19LET N=2: E$="DRE"
3140 IF A=20LET N=2: E$="REC"
3150 IF A=21LET N=2: E$="QUA"
3160 IF A=22LET N=3: E$="SEN"
3170 GOTO "PKTEIN"
3180 REM -----
3190 "MENUE3"
3200 REM -----
3210 PRINT " K-1 K-2 K-3 K-4 K-5 K-6"
3220 "INFO3" A=ASC INKEY$ : IF A=0GOTO 3220
3230 IF A=9GOTO "MENUE4"
3240 IF A=25GOTO "MENUE0"
3250 IF A<17OR A>22GOTO "INFO3"
3260 IF A=17LET N=1: E$="K-1"
3270 IF A=18LET N=2: E$="K-2"
3280 IF A=19LET N=2: E$="K-3"
3290 IF A=20LET N=3: E$="K-4"
3300 IF A=21LET N=3: E$="K-5"
3310 IF A=22LET N=2: E$="K-6"
3320 GOTO "PKTEIN"
```

Do not sale !

```
3330 REM -----
3340 "MENUE4"
3350 REM -----
3360 PRINT " B-1 B-2 B-3 E-1 E-2"
3370 "INFO4"A=ASC INKEY$ : IF A=0GOTO 3370
3380 IF A=9GOTO "MENUE5"
3390 IF A=25GOTO "MENUE0"
3400 IF A<17OR A>21GOTO "INFO4"
3410 IF A=17LET N=1:E$="B-1"
3420 IF A=18LET N=2:E$="B-2"
3430 IF A=19LET N=3:E$="B-3"
□ 3440 IF A=20LET N=1:E$="E-1"
□ 3450 IF A=21LET N=1:E$="E-2"
3460 GOTO "PKTEIN"
3470 REM -----
3480 "MENUE5"
3490 REM -----
3500 PRINT " RAS TEX SON"
3510 "INFO5"A=ASC INKEY$ : IF A=0GOTO 3510
3520 IF A=9GOTO "MENUE10"
3530 IF A=25GOTO "MENUE0"
3540 ON A-16GOTO "RAS","TEX","SON"
3550 GOTO "INFO5"
3560 REM -----
3570 "TEX"
3580 REM -----
3590 M=7
3600 INPUT "TEXT EINGEBEN : ";TE$(0)
3610 IF D$=""GLCURSOR (U,V):LPRINT TE$(0)
3620 IF D$<>"LPRINT "M";U;",";V:LPRINT "P";TE$(0)
3630 GOTO "MENUE00"
△ 3640 REM -----
3650 "RAS"
3660 REM -----
3670 INPUT "RASTERGROSSE ";RA
3680 D=INT ((H-1)/RA):A=0:O=B
3690 FOR I=1TO D
3700 IF D$=""LINE (A,I*RA)-(O,I*RA),L
3710 IF D$<>"LPRINT "M";A;",";I*RA:LPRINT "D";O;",";I*RA
3720 Q=A:A=O:O=Q
3730 NEXT I
3740 D=INT ((B-1)/RA):A=H:O=0
3750 FOR I=1TO D
3760 IF D$=""LINE (I*RA,A)-(I*RA,O),L
△ 3770 IF D$<>"LPRINT "M";I*RA;",";A:LPRINT "D";I*RA;",";O
```

Do not sale !

```
Δ 3780 Q=A:A=0:0=Q
| 3790 NEXT I
Δ 3800 GOTO "MENUE00"
O 3810 REM -----
  3820 "SON"
  3830 REM -----
  3840 M=8
  3850 CLS :GCURSOR 12:GPRINT "3E222E223E":GCURSOR 36:GPRINT
"1C222E221C":GCURSOR 60:GPRINT "08083E0808"
  3860 GCURSOR 84:GPRINT "2214081422":GCURSOR 108:GPRINT "2A1
C3E1C2A"
  3870 "INFO-SON"N=ASC INKEY$ :IF N=0GOTO 3870
  3880 IF N<17OR N>21GOTO "INFO-SON"
  3890 GOSUB "SOZEI":GOTO "MENUE00"
  3900 REM -----
  3910 "SOZEI"
  3920 REM -----
  3930 RESTORE "SON-"+CHR$(N+32)
  3940 "SO-1"
  3950 READ I:IF I=999RETURN
  3960 IF I>0GOTO "SO-2"
  3970 I=-I
  3980 READ J,K
  3990 IF D$=""RLINE (J*S,K*S)-(0,0),9
  4000 IF D$<>""LPRINT "R";J*S;",";K*S
  4010 "SO-2"
  4020 FOR Q=1TO I
  4030 READ J,K
  4040 IF D$=""RLINE -(J*S,K*S),L
  4050 IF D$<>""LPRINT "J";J*S;",";K*S
  4060 NEXT Q
  4070 GOTO "SO-1"
  4080 REM -----
  4090 "SON-1"DATA 6,0,2,-2,0,0,-4,4,0,0,4,-2,0,999
  4100 "SON-2"DATA 10,0,2,-1,0,-1,-1,0,-2,1,-1,2,0,1,1,0,2,-1
,1,-1,0,999
  4110 "SON-3"DATA -1,0,2,0,-4,-1,-2,2,4,0,999
  4120 "SON-4"DATA -1,-2,-2,4,4,-1,-4,0,4,-4,999
O 4130 "SON-5"DATA -1,-2,2,4,-4,-1,-4,0,4,4,-1,-2,0,0,-4,-1,-
2,2,4,0,999
  4140 REM -----
  4150 "MENUE10"
  4160 REM -----
  4170 PRINT " TES ZEI          SAV END"
```

Do not sale !

```
4180 "INFO10"A=ASC INKEY$ :IF A=0GOTO 4180
4190 IF A=9GOTO "MENUE0"
4200 IF A=25GOTO "MENUE0"
4210 IF A=17LET ST=1:GOTO "ZEICHNE"
4220 IF A=18LET ST=0:GOTO "ZEICHNE"
4230 IF A=21PRINT #I1,I2,I3,I4,SP$(*),SP$(*):GOTO "MENUE0"
4240 IF A=22AND D$=""GLCURSOR (0,0):TEXT :END
4250 IF A=22AND D$<>"LPRINT "M0,0":LPRINT "A":SETDEV :END

4260 GOTO "INFO10"
4270 "ZEICHNE"
4280 IF D$=""GLCURSOR (0,-H-10):SORGN
4290 IF D$<>"LPRINT "M0,";-H-10:LPRINT "I"
4300 J1=0:J2=0:J3=0:J4=0
4310 "TE0"IF J1=I1AND J2=I2GOTO "MENUE00"
4320 PRINT "ZEICHNEN"
4330 K1=J1:K2=J2:K3=J3:K4=J4
4340 REM -----
4350 M=ASC SP$(J1,J2):GOSUB "TE-SP$"
4360 C=ASC SP$(J1,J2):GOSUB "TE-SP$"
4370 IF D$=""COLOR C
4380 IF D$<>"LPRINT CHR$(27)+CHR$(830+C)
4390 IF M=10R M=20R M=70R M=80LET N=ASC SP$(J1,J2):GOSUB "TE-SP$"
4400 IF M=70R M=80LET F=N
4410 IF M<>7GOSUB "TE-L"
4420 IF M=70R M=80GOSUB "TE-S"
4430 IF M=7LET R=ASC SP$(J1,J2):GOSUB "TE-SP$"
4440 IF M<>1AND M<>2LET N=1
4450 FOR I=0TO N-1:GOSUB "TE-P":U(I)=Q:GOSUB "TE-P":U(I)=Q:
NEXT I
4460 IF M=30R M=40R M=50R M=60LET RA=SP(J3,J4):GOSUB "TE-SP"
:W=SP(J3,J4):GOSUB "TE-SP"
4470 IF M=50R M=60LET RB=SP(J3,J4):GOSUB "TE-SP":AL=SP(J3,J4
):GOSUB "TE-SP"
4480 IF M=30R M=50LET W1=0:W2=360
4490 IF M=40R M=60LET W1=SP(J3,J4):GOSUB "TE-SP":W2=SP(J3,J4
):GOSUB "TE-SP"
4500 REM -----
4510 ON MGOSUB "LINIE","LINIE","KREIS","KREIS","ELLIPSE","E
LLIPSE","TESTTEXT","TESTSO
4520 GOTO "TEST"
4530 REM -----
4540 "TESTTEXT"
4550 REM -----
```

Do not sale !

```
4560 IF D$="" ROTATE R
4570 IF D$<>"" LPRINT "Q";R
4580 TE$(0)="" :FOR I=1 TO F:TE$(0)=TE$(0)+SP$(J1,J2):GOSUB "
TE-SP$":NEXT I
4590 IF D$="" GLCURSOR (U(0),V(0)):LPRINT TE$(0)
4600 IF D$<>"" LPRINT "M";U(0);", ";V(0):LPRINT "P";TE$(0)
4610 RETURN
4620 REM -----
4630 "TESTSO"
4640 REM -----
4650 N=F
4660 IF D$="" GLCURSOR (U(0),V(0))
4670 IF D$<>"" LPRINT "M";U(0);", ";V(0)
4680 GOSUB "SOZEI"
4690 RETURN
4700 REM -----
4710 "TEST"
4720 REM -----
4730 IF ST=0 GOTO "TE0"
4740 PRINT " LOE WEI END"
4750 "TEST-INFO" A=ASC INKEY$ :IF A=0 GOTO 4750
4760 IF A=18 CLS :GOTO "TE0"
4770 IF A=19 GOTO "MENUE00"
4780 IF A<>17 GOTO "TEST-INFO"
4790 PRINT "LOESCHEN"
4800 REM -----
4810 REM SP$(*) LOESCHEN
4820 REM -----
4830 IF J1=K1 AND J2=K2 GOTO "LOE3"
4840 I=K1:J=K2
4850 IF J1=I1 AND J2=I2 GOTO "LOE2"
4860 "LOE1" SP$(I,J)=SP$(J1,J2)
4870 J=J+1:IF J>D2 LET J=0:I=I+1
4880 J2=J2+1:IF J2>D2 LET J2=0:J1=J1+1
4890 IF J1<>I1 OR J2<>I2 GOTO "LOE1"
4900 "LOE2" J1=K1:J2=K2:I1=I:I2=J
4910 REM -----
4920 "LOE3"
4930 REM -----
4940 REM SP$(*) LOESCHEN
4950 REM -----
4960 IF J3=K3 AND J4=K4 GOTO "TE0"
4970 I=K3:J=K4
4980 IF J3=I3 AND J4=I4 GOTO "LOE5"
```

Do not sale !

```
4990 "LOE4"SP(I,J)=SP(J3,J4)
5000 J=J+1:IF J>D4LET J=0:I=I+1
5010 J4=J4+1:IF J4>D4LET J4=0:J3=J3+1
5020 IF J3<>I3OR J4<>I4GOTO "LOE4"
5030 "LOE5"J3=K3:J4=K4:I3=I:I4=J
5040 GOTO "TE0"
5050 REM -----
5060 "TE-L"
5070 REM -----
5080 L=ASC SP$(J1,J2)
5090 IF D$<>"L"PRINT "L";L
5100 GOTO "TE-SP$"
5110 REM -----
5120 "TE-S" S=ASC SP$(J1,J2)
5130 IF D$=""CSIZE S
5140 IF D$<>"L"PRINT CHR$(27)+"?"+CHR$(860+S)
5150 GOTO "TE-SP$"
5160 REM -----
5170 "TE-SP$"J2=J2+1:IF J2>D2LET J2=0:J1=J1+1
5180 RETURN
5190 REM -----
5200 "TE-SP"J4=J4+1:IF J4>D4LET J4=0:J3=J3+1
5210 RETURN
5220 REM -----
5230 "TE-P" P1=ASC SP$(J1,J2):U0=0:IF P1>=2^7LET P1=P1-2^7:U
0=1
5240 Q=P1*256:GOSUB "TE-SP$"
5250 Q=Q+ASC SP$(J1,J2):GOSUB "TE-SP$"
5260 IF U0=1LET Q=-Q
5270 RETURN
5280 REM -----
5290 "L-N"
5300 REM -----
5310 M=1:GOSUB "LINIE":GOTO "MENUE00"
5320 REM -----
5330 "U-N"
5340 REM -----
5350 M=2:GOSUB "LINIE":GOTO "MENUE00"
5360 REM -----
5370 "LINIE"
5380 REM -----
5390 IF D$<>"G" GOTO "LI1"
5400 GLCURSOR (U(0),U(0))
5410 FOR I=0TO N-1:LINE -(U(I),U(I)),L:NEXT I
5420 IF M=2LINE -(U(0),U(0)),L
5430 RETURN
```

Do not sale !

```
5440 "LI1"LPRINT "M";U(0);", ";U(0)
5450 FOR I=1TO N-1:LPRINT "D";U(I);", ";U(I):NEXT I
5460 IF M=2LPRINT "D";U(0);", ";U(0)
5470 RETURN
5480 REM -----
5490 "DRE"
5500 REM -----
5510 INPUT "WINKEL 1 ? ";W1:IF W1<=0OR W1>=180GOTO 5510
5520 INPUT "WINKEL 2 ?";W2:IF W2<=0OR W2>=180GOTO 5520
5530 IF (W1+W2)>=180GOTO "DRE"
5540 W3=180-W1-W2
5550 J=1:GOSUB "TRANS"
5560 UU(2)=UU(1)*SIN W2/SIN W3*COS W1
5570 VV(2)=UU(1)*SIN W2/SIN W3*SIN W1
5580 J=2:GOSUB "UU->U"
5590 N=3:GOTO "U-N"
5600 REM -----
5610 "REC"
5620 REM -----
5630 INPUT "ABSTAND ? ";AB
5640 J=1:GOSUB "TRANS"
5650 UU(2)=UU(1):VV(2)=AB
5660 UU(3)=0:VV(3)=AB
5670 J=2:GOSUB "UU->U"
5680 J=3:GOSUB "UU->U"
5690 N=4:GOTO "U-N"
5700 REM -----
5710 "QUA"
5720 REM -----
5730 U(2)=U(1):V(2)=V(1)
5740 J=2:GOSUB "TRANS"
5750 UU(1)=UU(2)/2:VV(1)=-UU(1)
5760 UU(3)=UU(1):VV(3)=UU(3)
5770 J=1:GOSUB "UU->U"
5780 J=3:GOSUB "UU->U"
5790 N=4:GOTO "U-N"
5800 REM -----
5810 "SEN"
5820 REM -----
5830 J=1:GOSUB "TRANS"
5840 J=2:GOSUB "U->UU"
5850 UU(0)=UU(2)
5860 VV(0)=0
5870 J=0:GOSUB "UU->U"
5880 U(1)=U(2):V(1)=V(2)
5890 N=2:GOTO "L-N"
```

Do not sale !


```
5900 REM -----
5910 "K-1"
5920 REM -----
5930 INPUT "RADIUS ? ";RA:GOTO "KR"
5940 REM -----
5950 "K-2"
5960 REM -----
5970 RA=SQR ((U(1)-U(0))^2+(V(1)-V(0))^2)/2:U(0)=(U(0)+U(1)
)/2
5980 V(0)=(V(0)+V(1))/2:GOTO "KR"
5990 REM -----
6000 "K-3"
6010 REM -----
6020 RA=SQR ((U(1)-U(0))^2+(V(1)-V(0))^2):GOTO "KR"
6030 REM -----
6040 "K-4"
6050 REM -----
6060 J=1:GOSUB "TRANS":J=2:GOSUB "U->UU"
6070 UU(0)=UU(1)/2
6080 VV(0)=(UU(2)^2-UU(1)*UU(2)+UU(2)^2)/(2*UU(2))
6090 RA=SQR (UU(0)^2+VV(0)^2)
6100 J=0:GOSUB "UU->U"
6110 GOTO "KR"
6120 REM -----
6130 "K-5"
6140 REM -----
6150 J=1:GOSUB "TRANS":J=2:GOSUB "U->UU"
6160 RA=ABS VV(2)
6170 U(0)=U(2):V(0)=V(2)
6180 GOTO "KR"
6190 REM -----
6200 "K-6"
6210 REM -----
6220 INPUT "RADIUS ? ";RA
6230 J=1:GOSUB "TRANS"
6240 VV(0)=RA
6250 J=0:GOSUB "UU->U"
6260 GOTO "KR"
6270 REM -----
6280 "KR"
6290 REM -----
6300 W1=0:W2=360:M=3:GOSUB "KREIS":GOTO "MENUE00"
6310 REM -----
6320 "B-1"
6330 REM -----
```

Do not sale !

```
6340 INPUT "RADIUS ? ";RA:INPUT "WINKEL 1 ";W1:INPUT "WINKE
L 2 ";W2:GOTO "B0"
6350 REM -----
6360 "B-2"
6370 REM -----
6380 INPUT "RADIUS ? ";RA:J=1:GOSUB "TRANS"
6390 UU(2)=UU(1)/2:VV(2)=SQR (RA^2-UU(2)^2)
6400 J=2:GOSUB "UU->U"
6410 Q=U(0):U(0)=U(2):U(2)=U(1):U(1)=Q
6420 Q=U(0):U(0)=U(2):U(2)=U(1):U(1)=Q
6430 "B22"
6440 I=0:J=1:GOSUB "STEIGUNG":W1=AL
6450 I=0:J=2:GOSUB "STEIGUNG":W2=AL
6460 GOTO "B0"
6470 REM -----
6480 "B-3"
6490 REM -----
6500 RA=SQR ((U(1)-U(0))^2+(V(1)-V(0))^2)
6510 GOTO "B22"
6520 REM -----
6530 "B0"
6540 REM -----
6550 M=4:GOSUB "KREIS":GOTO "MENUE00"
6560 REM -----
6570 "KREIS"
6580 REM -----
6590 IF W1>W2LET W2=W2+360:GOTO 6590
6600 PH=W1
6610 X=U(0)+RA*COS PH
6620 Y=U(0)+RA*SIN PH
6630 IF D$=""GLCURSOR (X,Y)
6640 IF D$<>""LPRINT "M";X;",";Y
6650 "NEXT PH"
6660 PH=PH+W
6670 IF PH>W2LET PH=W2
6680 X=U(0)+RA*COS PH
6690 Y=U(0)+RA*SIN PH
6700 IF D$=""LINE -(X,Y),L
6710 IF D$<>""LPRINT "D";X;",";Y
6720 IF PH<W2GOTO "NEXT PH"
6730 RETURN
□ 6740 REM -----
  | 6750 "E-1"
  | 6760 REM -----
□ 6770 M=5
```

Do not sale !

```
□ 6780 "E11"
6790 INPUT "1. HALBACHSE = ? ";RA
6800 INPUT "2. HALBACHSE = ? ";RB
6810 INPUT "DREHWINKEL = ? ";AL
6820 IF M=5LET W1=0:W2=360
6830 IF M=6INPUT "WINKEL 1 ? ";W1:INPUT "WINKEL 2 ?";W2
6840 GOSUB "ELLIPSE":GOTO "MENUE00"
6850 REM -----
6860 "E-2"
6870 REM -----
6880 M=6
6890 GOTO "E11"
6900 REM -----
6910 "ELLIPSE"
6920 REM -----
6930 IF W2<W1LET W2=W2+360:GOTO 6930
6940 PH=W1
6950 UU(0)=0:VV(0)=0
6960 UU(1)=RA*COS PH
6970 VV(1)=RB*SIN PH
6980 J=1:GOSUB "UU->U"
6990 IF D$=""GLCURSOR (U(1),V(1))
7000 IF D$<>""LPRINT "M";U(1);", ";V(1)
7010 II=360/W
7020 FOR I=0TO II
7030 PH=PH+W
7040 IF PH>W2LET PH=W2:I=II
7050 UU(1)=RA*COS PH
7060 VV(1)=RB*SIN PH
7070 J=1:GOSUB "UU->U"
7080 IF D$=""LINE -(U(1),V(1)),L
7090 IF D$<>""LPRINT "D";U(1);", ";V(1)
7100 NEXT I
□ 7110 RETURN
```

Do not sale !

Hier noch einige weitere Hinweise:

Beim Speichern einer Grundkonstruktion müssen je nach Art der Konstruktion folgende Angaben gespeichert werden:

Name der Grundkonstruktion	Nummer der Grundkonstruktion M	Farbe C	Linientyp L	Größe S	Schreibrichtung R	Zahl der Punkte N	N Punktkoordinatenpaare	Radius RA	Radius RB	Winkelinkrement W	Anfangswinkel W1	Endwinkel W2	Drehrichtung AL	Textlänge IL	Text TE\$	Nummer des Sonderzeichens N	Hinweise
L-N	1	x	x			x	x										
V-N	2	x	x			x	x										
K	3	x	x				x	x		x							K-1 ... K-6
B	4	x	x				x	x		x	x	x					B-1 ... B-3
E-1	5	x	x				x	x	x	x				x			
E-2	6	x	x				x	x	x	x	x	x	x				
TEX	7	x		x	x		x							x	x		
SON	8	x	x	x			x										x

Es kommt nun darauf an, diese Zahlen möglichst platzsparend zu speichern. Man könnte zwar jede Angabe in einem eigenen Feld-element speichern, dann würden aber pro Zahl 8 Bytes verbraucht werden. In einem Byte kann man aber bereits eine Integerzahl

Do not sale !

zwischen 0 und 255 speichern. Viele Zahlen, die wir abspeichern wollen, lassen sich also in einem Byte unterbringen. Die Punktkoordinaten verbrauchen zwei Bytes, da sie bis zu 999 gehen können. (Die Winkel und Radiusangaben werden normal in 8 Bytes als Fließkommazahl gespeichert, da sie Nachkommastellen haben können.) Felder, die genau ein Byte lang sind, lassen sich beim PC-1500 (A) nur als Stringvariablen definieren: DIM SP\$(10)*1. Die Farbe C wird dann z.B. abgespeichert als

$$SP$(5) = CHR$ C$$

und aus dem Speicher genommen mit

$$C = ASC SP$(5)$$

Leider dürfen eindimensionale Felder (zumindest beim PC-1500 (A)) nicht mehr als 256 Elemente besitzen. Deswegen sind die Felder SP\$ und SP zweidimensional definiert. Das wirft natürlich Probleme beim Belegen der Felder auf (Zeile 2880 und 2910), da die Felder immer zeilenweise belegt werden. Die Dimensionierung der Felder SP und SP\$ erfolgt in Zeile 1370 von GRADOR. Bitte ändern Sie diese Zeile entsprechend um, wenn die Felder für Ihre Belange zu groß oder zu klein sind.

LITERATUR:

- : POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung. 1. Aufl.; Würzburg 1984
- : BROWN, Frederick J.: Coordinate Geometry for Computer Graphics. North Hollywood 1969
- : CHASEN, Sylvan H.: Geometric Principles and Procedures for Computer Graphic Applications. Englewood Cliffs 1978

Do not sale !

: ROGERS, David F. / ADAMS, J. Alan: Mathematical Elements
for Computer Graphics. New York 1976

: BRONSTEIN, I. N. / SEMENDJAJEW, K. A.: Taschenbuch der
Mathematik. Thun 1980

PROGRAMM-LISTINGS FÜR PC-1500 (A):

: CAD aus den Taschencomputer. In: Zeitschrift CHIP, Nr. 9/1983

: QUICK-DESIGN. In: Zeitschrift Computer persönlich, Nr. 19/1984
(*Ein interessanter Grafik-Editor, ähnlich
dem Programm GRADOR.*)

Do not sale !

4 KREISDIAGRAMME

Der eben beschriebene Grafik-Editor ermöglicht das Zeichnen beliebiger Bilder. Diese große Flexibilität erkaufte man freilich durch zwei Nachteile:

- : GRADOR ist sehr lang
- : GRADOR ist relativ langsam.

Deswegen ist es zweckmäßig, für spezielle und immer wieder vorkommende Anwendungen eigene Programme zu schreiben. Ein solches Problem ist die Darstellung von Ergebnissen in Kreisdiagrammen.

In Kreisdiagrammen kann man beispielsweise die Ergebnisse von Wahlen darstellen:

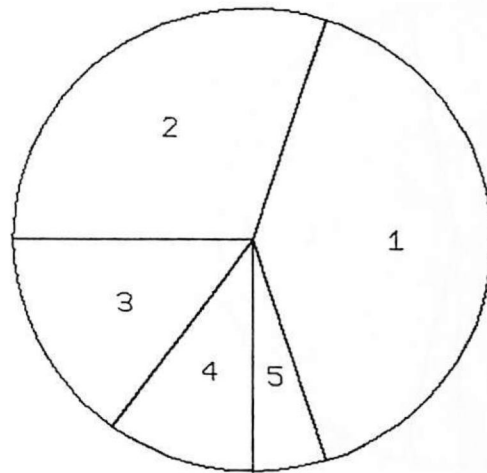
Bei einer Wahl erhielt

<i>Partei A</i>	<i>40 % der Stimmen</i>
<i>Partei B</i>	<i>30 % der Stimmen</i>
<i>Partei C</i>	<i>15 % der Stimmen</i>
<i>Partei D</i>	<i>10 % der Stimmen</i>
<i>sonstige</i>	<i>5 % der Stimmen.</i>

Diese Stimmenverteilung soll in einem Kreisdiagramm dargestellt werden. Dazu muß im Prinzip fünfmal das Kreisbogenunterprogramm von GRADOR aufgerufen werden, um für jede Partei ein Kreissegment zu zeichnen. Die Größe des Kreissegments richtet sich dabei nach der Zahl der für eine Partei abgegebenen Stimmen. Partei A erhält z.B. ein Segment mit $40\% \text{ von } 360^\circ = 144^\circ$ des Vollkreises.

Das im folgenden beschriebene Programm ist ein einfaches Beispiel dafür, wie man Kreisdiagramme erstellen kann. Durch die Anwendung dieses Programms erhält man für unser Beispiel folgendes Bild:

Do not sale !



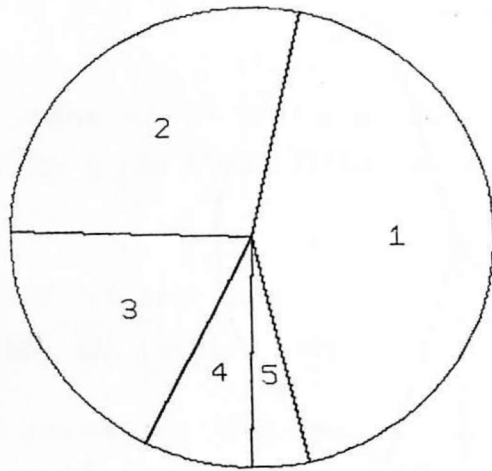
1	A	40.0
2	B	30.0
3	C	15.0
4	D	10.0
5	sonstige	5.0

Zur Anwendung des Programms braucht man aber die darzustellenden Größen nicht zu relativieren. Man kann (um bei unserem Beispiel zu bleiben) die Ergebnisse einer Wahl auch durch die Eingabe der abgegebenen Stimmen darstellen.

<i>Partei A</i>	<i>erhielt</i>	<i>3842</i>	<i>Stimmen</i>
<i>Partei B</i>	<i>erhielt</i>	<i>2523</i>	<i>Stimmen</i>
<i>Partei C</i>	<i>erhielt</i>	<i>1615</i>	<i>Stimmen</i>
<i>Partei D</i>	<i>erhielt</i>	<i>649</i>	<i>Stimmen</i>
<i>sonstige</i>	<i>erhielten</i>	<i>353</i>	<i>Stimmen</i>

Als Kreisdiagramm erhalten wir das auf der folgenden Seite dargestellte Bild.

Do not sale !



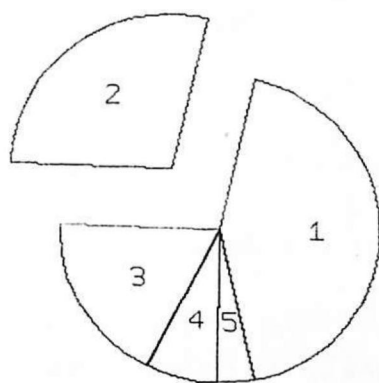
1	A	3842
2	B	2523
3	C	1615
4	D	649
5	sonstige	363

Vertreter der Partei B werden natürlich Wert auf die Hervorhebung ihrer Stimmen legen. Dazu kann man dem Programm sagen, daß es das Kreissegment der Partei B etwas aus dem Kreis lösen soll. Hierzu wird das Ergebnis der Partei B negativ eingegeben:

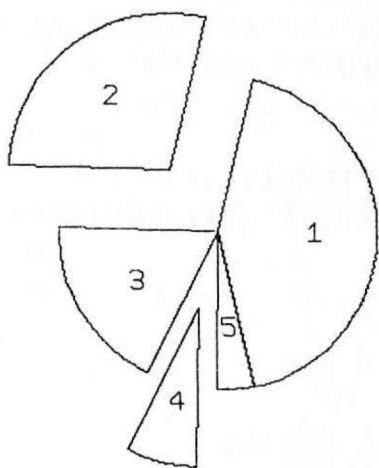
A	3842	Stimmen
B	-2523	Stimmen
C	1615	Stimmen
D	649	Stimmen
sonstige	353	Stimmen

Man kann auch mehrere Segmente ausdrücken, indem man mehrere Zahlen negativ eingibt. Beispiele dazu zeigen die nächsten beiden Bilder:

Do not sale !



1	A	3842
2	B	2523
3	C	1615
4	D	649
5	sonstige	353



1	A	3842
2	B	2523
3	C	1615
4	D	649
5	sonstige	353

Do not sale !

Doch nun das Programmlisting und einige Kommentare dazu:

```
3000 "K"
3010 REM -----
3020 REM  KREISDIAGRAMM
3030 REM -----
3040 WAIT 0:DEGREE
3050 DIM U(0),V(0)
3060 INPUT "Externer Drucker ? (J/N) ";D$
3070 IF D$<>"J"LET D$=""
3080 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PO
3090 IF D$<>" "CONSOLE 0,0
3100 IF D$=""GRAPH
3110 IF D$<>" "LPRINT CHR$(27)+"b"
3120 REM -----
3130 REM  EINGABE DER DATEN
3140 REM -----
3150 S=2:INPUT "CSIZE ? ";S
3160 U$="##.#":INPUT "USING-Format ? ";U$
3170 C=0:INPUT "COLOR ? ";C
3180 IF D$=""CSIZE S:COLOR C
3190 IF D$<>" "LPRINT CHR$(27)+CHR$(&30+C)
3200 IF D$<>" "LPRINT CHR$(27)+"?"+CHR$(&60+S)
3210 INPUT "Anzahl ? ";N:DIM P(N),P$(N)
3220 SU=0
3230 FOR I=1TO N
3240 CLS :INPUT "Name ? ";P$(I)
3250 PRINT "Wert von ";P$(I);" ";:INPUT P(I)
3260 SU=SU+ABS P(I)
3270 NEXT I
3280 CLS
3290 R=50:INPUT "Radius ? ";R
3300 REM -----
3310 REM  MAX. LAENGE DES TEXTES
3320 REM -----
3330 IL=0
3340 FOR I=1TO N
3350 IF LEN P$(I)>ILLET IL=LEN P$(I)
3360 NEXT I
3370 REM -----
3380 REM  KREISDIAGRAMM ZEICHNEN
3390 REM -----
3400 IF D$=""GLCURSOR (1.5*R,-1.5*R):SORGN
3410 IF D$<>" "LPRINT "M";1.5*R;" ";-1.5*R:LPRINT "I"
```

Do not sale !

```
3420 FOR I=1TO N
3430 WI=ABS P(I)*360/SU
3440 IF I=1LET W1=-WI/2
3450 IF I<>1LET W1=W2
3460 W2=W1+WI
3470 U(0)=0:V(0)=0
3480 REM -----
3490 REM WENN P(I) NEGATIV, DANN SEGMENT AUSRUECKEN
3500 REM -----
3510 IF P(I)<0LET U(0)=.5*R*COS (W1+WI/2)
3520 IF P(I)<0LET V(0)=.5*R*SIN (W1+WI/2)
3530 GOSUB "KREIS"
3540 REM -----
3550 REM BESTIMME ORT FUER SEGMENTBESCHRIFTUNG
3560 REM -----
3570 U=U(0)+.6*R*COS (W1+WI/2):V=V(0)+.6*R*SIN (W1+WI/2)
3580 IF D$=""GLCURSOR (U,V):RLINE -(-S*8,-S*3),9
3590 IF D$<>""LPRINT "M";U;",";V:LPRINT "R";-S*8;",";"-S*3
3600 IF D$=""LPRINT I
3610 IF D$<>""LPRINT "P";I
3620 NEXT I
3630 REM -----
3640 REM TEXTAUSGABE
3650 REM -----
3660 IF D$=""GLCURSOR (0,-1.5*R):TEXT :CSIZE S
3670 IF D$<>""LPRINT "M0,";-1.5*R:LPRINT "A"
3680 IF D$<>""LPRINT CHR$ (27)+"?"+CHR$ (&60+S)
3690 FOR I=1TO N
3700 LPRINT USING "###";I;
3710 LPRINT TAB 5;P$(I);
3720 USING U$
3730 LPRINT TAB 7+IL;ABS P(I)
3740 NEXT I
3750 IF D$<>""SETDEV
3760 END
3770 REM -----
3780 REM UP ZUM KREISZEICHNEN
3790 REM -----
3800 "KREIS"
3810 PH=W1
3820 U=U(0)+R*COS PH
3830 V=V(0)+R*SIN PH
3840 IF D$=""GLCURSOR (U(0),V(0)):LINE -(U,V)
3850 IF D$<>""LPRINT "M";U(0);",";V(0):LPRINT "D";U;",";V
3860 REM -----
3870 "KR"PH=PH+5
3880 IF PH>W2LET PH=W2
```

Do not sale !

```
3890 U=U<0>+R*COS PH
3900 U=U<0>+R*SIN PH
3910 IF D$=""LINE -(U,U),0
3920 IF D$<>""LPRINT "D";U;",";U
3930 IF PH<>W2GOTO "KR"
3940 IF D$=""LINE -(U<0>,U<0>)
3950 IF D$<>""LPRINT "D";U<0>;",";U<0>
3960 RETURN
```

Man sieht, daß bei vielen INPUT-Anweisungen eine Voreinstellung realisiert wurde:

```
C = 0 : INPUT "COLOR ? ";C
```

Wenn nun die INPUT-Eingabe mit einer leeren Eingabe (ENTER) beantwortet wird, dann wird die Voreinstellung wirksam.

Die Felder P(N) und P\$(N) für die Zahlenwerte und Namen der Gruppen werden erst dimensioniert, wenn die Anzahl bekannt ist (Zeile 3210). In den Zeilen 3300 bis 3360 wird die maximale Länge der Namen P\$(I) bestimmt, um später beim Ausdrucken der Namen eine "schöne" Darstellung zu erhalten. Um auch die Zahlen tabellarisch ausgeben zu können, muß in Zeile 3160 das USING-Format der Ausgabe eingegeben werden.

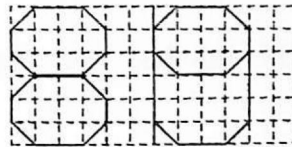
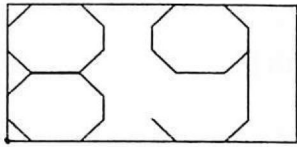
Kreissegmente verlaufen vom Winkel W1 bis zum Winkel W2. Das erste Kreissegment wird immer so angeordnet, daß $W1 = -W2$ ist (Zeile 3440). Alle weiteren Segmente schließen immer an das vorhergehende an (Zeile 3450). Zum Hervorheben einzelner Kreissegmente wird der Kreismittelpunkt um den halben Kreisradius verschoben. Die Richtung W des Verschiebens ist die Winkelhalbierende des Segments:
 $W = W1 + W2 / 2$. Das Unterprogramm "KREIS" unterscheidet sich vom GRADOR-Kreis-UP dadurch, daß die seitlichen Begrenzungen des Kreissegments mitgezeichnet werden. Es ist eine feste Winkelschrittweite von 5° vorgegeben.

Die Beschriftung der Segmente erfolgt "optisch schön" etwa in der Mitte der Fläche. Gesucht wird zunächst der Punkt des Segments,

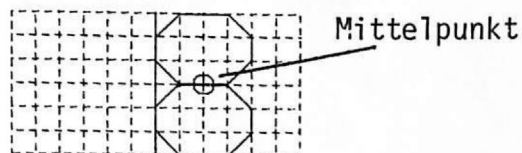
Do not sale !

der auf der Segmentwinkelhalbierenden im Abstand $\varnothing.6 * R$ vom Kreismittelpunkt liegt (Zeile 357 \varnothing). Von dort (das soll der Mittelpunkt der Segmentnummer sein) geht der Stift in die linke untere Ecke der Nummer. Dazu muß folgendes berücksichtigt werden:

Die Größe von Zeichen ist immer $4 * 6$ Einheiten. Dazu kommen zwei



Einheiten als Abstand zum nächsten Zeichen. Weiterhin muß beachtet werden, daß zu einer Zahl immer ein Vorzeichen gehört, auch wenn das positive Vorzeichen nicht geplottet wird. Zum Plotten einer einstelligen Zahl muß man also vom Mittelpunkt der Zahl aus um acht Einheiten nach links und um drei nach unten gehen. Diese Zahlen 8 und 3 müssen noch mit dem Faktor der Schriftgröße (CSIZE) multipliziert werden (Zeile 358 \varnothing).



Noch ein letzter Hinweis: Verstehen Sie das vorliegende Kreisdiagrammprogramm bitte nicht als das Super-Programm zum Zeichnen von Kreisdiagrammen. Es gibt bessere. In diesem Buch geht es nicht um das Vorstellen von Spitzenprogrammen. Es geht vielmehr darum, ein Grundgerüst für spezielle Programmanwendungen zu erklären, das leicht verständlich und einfach auf eigene Bedürfnisse erweiterbar ist.

Do not sale !

LITERATUR:

: POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung.
1. Aufl.; Würzburg 1984

PROGRAMM-KASSETTEN FÜR PC-1500 (A):

(Erhältlich bei der Fischel GmbH, Berlin)

: SUPER GRAPH *(Programm zur Darstellung grafischer
Übersichten. Möglich sind zwei- und
dreidimensionale Balkendiagramme, Kreis-
und Liniendiagramme.)*

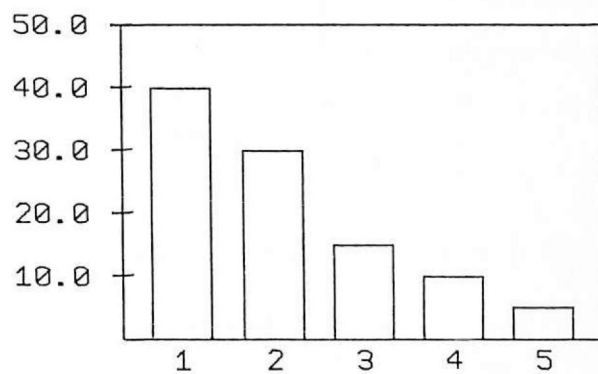
: CALC/PLOT *(Programm zur Bearbeitung von Tabellen.
Die Ergebnisse der Tabellenberechnung
können als zwei- und dreidimensionale
Balkendiagramme, als Kreis- und als
Liniendiagramme dargestellt werden.
Zur Tabellenberechnung können auch
Formeln eingegeben werden.)*

Do not sale !

5 BALKENDIAGRAMME

Analog zu den Kreisdiagrammen kann man Ergebnisse auch in Form von Balkendiagrammen darstellen. Wir wollen die Wahlergebnisse vom letzten Abschnitt darstellen:

Partei A : 40 %
Partei B : 30 %
Partei C : 15 %
Partei D : 10 %
sonstige : 5 %



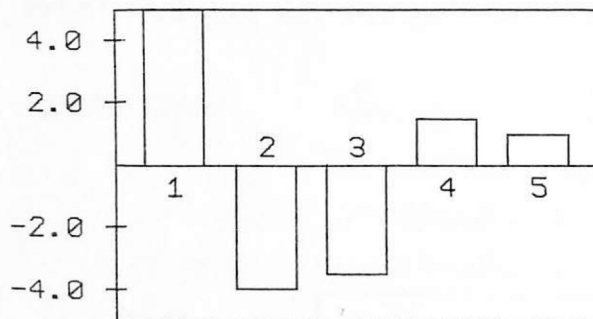
1	A	40.0
2	B	30.0
3	C	15.0
4	D	10.0
5	sonstige	5.0

Beim Balkendiagramm wird keine Relativierung der Ergebnisse wie beim Kreisdiagramm durchgeführt. Dafür können Balkendiagramme auch als Abweichungsdiagramme verwendet werden:

Do not sale !

Im Vergleich zur letzten Wahl hatte

Partei A 5.0 % Stimmengewinn
Partei B 4.0 % Stimmenverlust
Partei C 3.5 % Stimmenverlust
Partei D 1.5 % Stimmengewinn
sonstige 1.0 % Stimmengewinn



1	A	5.0
2	B	4.0
3	C	3.5
4	D	1.5
5	sonstige	1.0

Nachfolgend das Programmlisting:

```
4000 "B"  
4005 REM -----  
4010 REM BALKENDIAGRAMM  
4015 REM -----  
4020 WAIT 0  
4025 INPUT "Externer Drucker ? (J/N) ";D$  
4030 IF D$<>"J"LET D$=""  
4035 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV P0  
4040 IF D$<>" "CONSOLE 0,0
```

Do not sale !

```
4045 IF D$=""GRAPH
4050 IF D$<>""LPRINT CHR$ (27)+"b"
4055 REM -----
4060 REM  EINGABE DER DATEN
4065 REM -----
4070 S=2:INPUT "CSIZE ? ";S
4075 C=0:INPUT "COLOR ? ";C
4080 IF D$=""CSIZE S:COLOR C
4085 IF D$<>""LPRINT CHR$ (27)+"?"+CHR$ (&60+S)
4090 IF D$<>""LPRINT CHR$ (27)+CHR$ (&30+C)
4095 INPUT "Anzahl ? ";N:DIM P(N),P$(N)
4100 FOR I=1TO N
4105 CLS :INPUT "Name ? ";P$(I)
4110 PRINT "Wert von ";P$(I);" ";:INPUT P(I)
4115 NEXT I
4120 CLS
4125 B=100:INPUT "Breite ? ";B
4130 H=100:INPUT "Hoehe ? ";H
4135 REM -----
4140 "Bminmax"
4145 REM -----
4150 YK=0:INPUT "Min. Wert ? ";YK:IF YK>0GOTO 4150
4155 YG=10:INPUT "Max. Wert ? ";YG
4160 SW=1:INPUT "Schrittweite ? ";SW
4165 FOR I=1TO N
4170 IF P(I)<YKWAIT :PRINT "Wert kleiner Y-min":WAIT 0:GOTO
"Bminmax"
4175 IF P(I)>YGWAIT :PRINT "Wert groesser Y-max":WAIT 0:GOT
O "Bminmax"
4180 NEXT I
4185 U$="###.#":INPUT "USING-Format ? ";U$
4190 US=LEN U$
4195 REM -----
4200 REM  MAX. LAENGE DES TEXTES
4205 REM -----
4210 IL=0
4215 FOR I=1TO N
4220 IF LEN P$(I)>ILLET IL=LEN P$(I)
4225 NEXT I
4230 REM -----
4235 REM  RAHMEN ZEICHNEN
4240 REM -----
4245 IF D$<>""GOTO "B1"
4250 GLCURSOR (6*(US+2)*S,-H):SORGN
4255 LINE -(B,0)-(B,H)-(0,H)-(0,0)
4260 IF YK<0GLCURSOR (0,-YK/(YG-YK)*H):SORGN :LINE -(B,0)
4265 GLCURSOR (0,0)
```

Do not sale !

```
4270 GOTO "B2"
4275 REM -----
4280 "B1"
4285 REM -----
4290 LPRINT "M";6*(US+2)*S;",";-H:LPRINT "I"
4295 LPRINT "D";B;",";0;",";B;",";H;",";0;",";H;",";0;0"
4300 IF YK<0LPRINT "M0;",";-YK/(YG-YK)*H:LPRINT "I":LPRINT "D
";B;",";0"
4305 LPRINT "H"
4310 REM -----
4315 "B2"
4320 REM -----
4325 REM BALKENBREITE
4330 REM -----
4335 BB=B/(3*N+1)*2
4340 REM -----
4345 REM UMRECHNUNGSFAKTOR
4350 REM -----
4355 YF=H/(YG-YK)
4360 REM -----
4365 REM BESCHRIFTEN DER ACHSE
4370 REM -----
4375 Y=INT (YG/SW)*SW
4380 "B:NEXT I"
4385 IF Y=0LET Y=-SW:GOTO "B:NEXT I"
4390 IF Y<YK GOTO "B:ENDE I"
4395 U=Y*YF
4400 IF D$<>"GOTO "B3"
4405 GLCURSOR (0,U)
4410 RLINE (3*S,0)-(-6*S,0),0
4415 RLINE -(-(US+1)*S*6,-3*S),9
4420 USING U$
4425 LPRINT Y
4430 GOTO "B4"
4435 REM -----
4440 "B3"
4445 REM -----
4450 LPRINT "M0;";U
4455 LPRINT "R";3*S;",";0":LPRINT "J";-6*S;",";0"
4460 LPRINT "R";-(US+1)*S*6;",";";-3*S
4465 USING U$
4470 LPRINT "P";Y
```

Do not sale !

```
4475 REM -----
4480 "B4"
4485 REM -----
4490 USING
4495 Y=Y-SW
4500 GOTO "B:NEXT I"
4505 "B:ENDE I"
4510 REM -----
4515 REM BALKEN ZEICHNEN
4520 REM -----
4525 USING
4530 FOR I=1TO N
4535 IF D$<>"GOTO "B5"
4540 GLCURSOR (BB/2+(I-1)*BB*1.5,0)
4545 RLINE -(0,P(I)*YF)-(BB,0)-(0,-P(I)*YF)
4550 RLINE -(-BB/2-S*8,0),9
4555 IF P(I)<0RLINE -(0,3*S),9
4560 IF P(I)>=0RLINE -(0,-10*S),9
4565 LPRINT I
4570 GOTO "B6"
4575 REM -----
4580 "B5"
4585 REM -----
4590 LPRINT "M";BB/2+(I-1)*BB*1.5;"",0"
4595 LPRINT "J0,";P(I)*YF;"",";BB";",0,0,";-P(I)*YF
4600 LPRINT "R";-BB/2-S*8;"",0"
4605 IF P(I)<0LPRINT "R0,";3*S
4610 IF P(I)>=0LPRINT "R0,";-10*S
4615 LPRINT "P";I
4620 REM -----
4625 "B6"
4630 REM -----
4635 NEXT I
4640 REM -----
4645 REM TEXTAUSGABE
4650 REM -----
4655 IF D$=""GLCURSOR (0,-H/2+YK*YF):TEXT :CSIZE S
4660 IF D$<>"LPRINT "M0,";-H/2+YK*YF:LPRINT "A"
4665 IF D$<>"LPRINT CHR$ (27)+"?"+CHR$ (&60+S)
4670 FOR I=1TO N
4675 LPRINT USING "###";I;
4680 LPRINT TAB 5;P$(I);
4685 USING U$
4690 LPRINT TAB 7+IL;ABS P(I)
4695 NEXT I
4700 IF D$<>"SETDEV
4705 END
```

Do not sale !

Hier noch einige Hinweise:

Für die Skalierung der Daten muß der minimale Wert der Beschriftung der senkrechten Achse vorgegeben werden. Dieser Wert muß immer kleiner oder gleich Null sein (Zeile 4150).

Die Eingabe "Schrittweite" bezieht sich auf den Abstand der Beschriftung der senkrechten Achse.

Im Programm wird kontrolliert, ob die eingegebenen Daten den Wertebereich der senkrechten Achse überschreiten. Falls das der Fall ist, wird die neue Eingabe von minimalem und maximalem Wert verlangt.

Die senkrechte Achse wird in den Zeilen 4350 bis 4505 geplottet. YF ist der Umrechnungsfaktor zwischen eingegebenem y-Wert und gezeichnetem v-Wert. In der Variablen US ist die Länge des USING-Formates für die Achsenbeschriftung gespeichert. Damit wandert der Zeichenstift bei der Achsenbeschriftung weit genug nach links (Zeile 4415), um eine "schöne" Beschriftung zu ermöglichen.

In den Zeilen 4510 bis 4615 erfolgt das Plotten eines Balkens. BB ist die Balkenbreite, die je nach Zahl der Balken und der Bildbreite in Zeile 4335 berechnet wird. Falls der Balken nach oben gezeichnet wird, liegt die Balkenbeschriftung unterhalb, im umgekehrten Fall oberhalb der waagerechten Achse (Zeile 4555 bis 4560).

Die Tabelle am Ende des Programms ist analog zum Kreisdiagramm aufgebaut.

Do not sale !

LITERATUR:

: POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung.
1. Aufl.; Würzburg 1984

PROGRAMM-KASSETTEN FÜR PC-1500 (A):

(Erhältlich bei der Fischel GmbH, Berlin)

: SUPER GRAPH *(Programm zur Darstellung grafischer
Übersichten. Möglich sind zwei- und
dreidimensionale Balkendiagramme, Kreis-
und Liniendiagramme.)*

: CALC/PLOT *(Programm zur Bearbeitung von Tabellen.
Die Ergebnisse der Tabellenberechnung
können als zwei- und dreidimensionale
Balkendiagramme, als Kreis- und als
Liniendiagramme dargestellt werden.
Zur Tabellenberechnung können auch
Formeln eingegeben werden.)*

6 ZEICHNEN VON FUNKTIONEN

Neben den Kreis- und Balkendiagrammen ist auch das Plotten von Funktionen sehr wichtig. Kreis- und Balkendiagramme stellen einzelne Werte grafisch dar, während dies der Funktionszeichner kontinuierlich tut.

Ein einfaches Beispiel für einen Funktionszeichner zeigt das folgende Listing:

```
5000 Y=1/X
5004 RETURN
5008 REM -----
5012 REM FUNKTIONEN-PLOT
5016 REM -----
5020 "F"WAIT 0
5024 ON ERROR GOTO "ERROR"
5028 INPUT "EXTERNER DRUCKER ? (J/N) ";D$
5032 IF D$<>"J"LET D$=""
5036 IF D$<>" "SETCOM 1200,8,N,1:SETDEV PO:OUTSTAT 0
5040 IF D$<>" "CONSOLE 0,0
5044 IF D$=""GRAPH
5048 IF D$<>" "LPRINT CHR$ (27)+"b"
5052 REM -----
5056 REM DATENEINGABE
5060 REM -----
5064 XK=-5:INPUT "X-min ? ";XK:IF XK>0GOTO 5064
5068 XG=5:INPUT "X-max ? ";XG:IF XG<=XKOR XG<0GOTO 5068
5072 YK=-5:INPUT "Y-min ? ";YK:IF YK>0GOTO 5072
5076 YG=5:INPUT "Y-max ? ";YG:IF YG<=YKOR YG<0GOTO 5076
5080 XS=1:INPUT "X-Schritt ? ";XS
5084 YS=1:INPUT "Y-Schritt ? ";YS
5088 B=150:INPUT "Breite ? ";B
5092 H=150:INPUT "Hoehe ? ";H
5096 S=1:INPUT "CSIZE ? ";S
5100 U$="###.#":INPUT "USING-Format ? ";U$
5104 US=LEN U$
5108 IF D$=""COLOR 0:CSIZE S:GLCURSOR (6*(US+2)*S,0)
5112 IF D$<>" "LPRINT CHR$ (27)+CHR$ (&30)
5116 IF D$<>" "LPRINT CHR$ (27)+"?"+CHR$ (&60+S)
5120 IF D$<>" "LPRINT "M";6*(US+2)*S;"",0"
```

Do not sale !

```
5124 REM -----
5128 REM  KOORDINATENSYSTEM ZEICHNEN
5132 REM -----
5136 GOSUB "K-SYSTEM"
5140 REM -----
5144 REM  NEUE FARBE ?
5148 REM -----
5152 C=0:INPUT "COLOR ? ";C
5156 IF D$=""COLOR C
5160 IF D$<>""LPRINT CHR$ (27)+CHR$ (&30+C)
5164 SW=1
5168 X=XK
5172 U=UK
5176 REM -----
5180 REM  START DES FUNKTION-ZEICHNENS
5184 REM -----
5188 "F-2"FE=0:GOSUB 5000
5192 IF FE<>0LET U=U+SW:X=U/XF:GOTO "F-2"
5196 U=Y*YF
5200 UA=U:AA=0
5204 REM -----
5208 REM  WIRD DER U-BEREICH UEBER- ODER UNTERSCHRITTEN ?
5212 REM -----
5216 IF U>UGLET U=UG
5220 IF U<UKLET U=UK
5224 IF D$=""GLCURSOR (U,U)
5228 IF D$<>""LPRINT "M";INT U;",";INT U
5232 REM -----
5236 REM  MAX. X-WERT ERREICHT ?
5240 REM -----
5244 "F-3"IF X>=XGGOTO "F-ENDE"
5248 "F-4"
5252 REM -----
5256 REM  NAECHSTER X-WERT
5260 REM -----
5264 U=U+SW:IF U>UGLET SW=UG-U+SW:U=UG
5268 X=U/XF
5272 REM -----
5276 REM  BESTIMME Y(X)
5280 REM -----
5284 FE=0:GOSUB 5000
5288 IF FE<>0AND D$=""RLINE -(SW,0),9:SW=1:GOTO "F-3"
5292 IF FE<>0AND D$<>""LPRINT "R";SW;",";0":SW=1:GOTO "F-3"
5296 U=Y*YF
5300 REM -----
5304 REM  ABLEITUNG BERECHNEN
5308 REM -----
5312 A=(U-UA)/SW
5316 REM -----
```

Do not sale !


```
5320 REM KORREKTUR DER SCHRITTWEITE
5324 REM -----
5328 IF ABS (A-AA)<.1*ABS AAAND SW<16LET SW=SW*2
5332 IF ABS (A-AA)>.2*ABS AAAND SW>16LET U=U-SW:SW=SW/2:GOTO
"F-4"
5336 REM -----
5340 REM ZEICHNEN
5344 REM -----
5348 IF Y<=YGAND Y>=YKAND D$=""LINE -(U,U),0:GOTO "F-5"
5352 IF Y<=YGAND Y>=YKAND D$<>""LPRINT "D";INT U;",";INT U:
GOTO "F-5"
5356 REM -----
5360 REM SONDERFALL: KURVE AUSSERHALB DES WERTEBEREICHS
5364 REM -----
5368 IF V>UGAND V<UGAND VA>VKAND D$=""LINE -(U,UG),0:GOTO
"F-SW"
5372 IF V>UGAND V<UGAND VA>VKAND D$<>""LPRINT "D";INT U;",";
INT UG:GOTO "F-SW"
5376 IF V>UGAND D$=""GLCURSOR (U,UG):GOTO "F-SW"
5380 IF V>UGAND D$<>""LPRINT "M";INT U;",";INT UG:GOTO "F-S
W"
5384 IF V<VKAND VA>=VKAND V<UGAND D$=""LINE -(U,VK),0:GOTO
"F-SW"
5388 IF V<VKAND VA>=VKAND V<UGAND D$<>""LPRINT "D";INT U;",";
INT VK:GOTO "F-SW"
5392 IF V<VKAND D$=""GLCURSOR (U,VK):GOTO "F-SW"
5396 IF V<VKAND D$<>""LPRINT "M";INT U;",";INT VK:GOTO "F-S
W"
5400 REM -----
5404 "F-SW"SW=1
5408 REM -----
5412 "F-5"
5416 REM -----
5420 VA=U:AA=A
5424 GOTO "F-3"
5428 REM -----
5432 REM AUSGABE DER FUNKTION
5436 REM -----
5440 "F-ENDE"
5444 IF D$=""GLCURSOR (0,VK-10*S):TEXT :CSIZE S
5448 IF D$<>""LPRINT "M0, ";VK-10*S:LPRINT "A"
5452 IF D$<>""LPRINT CHR$ (27)+"?"+CHR$ (&60+S)
5456 LLIST 5000
5460 IF D$<>""SETDEV
5464 END
```

Do not sale !

```
5468 REM -----
5472 REM  UP ZUM ZEICHNEN EINES KOORDINATENSYSTEMS
5476 REM -----
5480 "K-SYSTEM"
5484 REM -----
5488  $XF=B/(XG-XK)$ 
5492  $UK=XK*XF$ 
5496  $UG=XG*XF$ 
5500  $YF=H/(YG-YK)$ 
5504  $UK=YK*YF$ 
5508  $UG=YG*YF$ 
5512 IF D$=""RLINE  $-(UK,UG),9$ :SORGN
5516 IF D$<>""LPRINT "R"; $-UK$ ;""; $-UG$ :LPRINT "I"
5520 REM -----
5524 REM  ZEICHNE X-ACHSE
5528 REM -----
5532 IF D$=""LINE  $(UK,0)-(UG,0),0$ 
5536 IF D$<>""LPRINT "M"; $UK$ ;"", $0$ ":LPRINT "D"; $UG$ ;"", $0$ "
5540  $X=INT(XG/XS)*XS$ : $U=X*XF$ 
5544 "KS-3"
5548 IF  $U=0$ GOTO "KS-1"
5552 IF  $UK<0$ GOTO "KS-2"
5556 IF D$<>""GOTO "F-6"
5560 GLCURSOR  $(U,0)$ 
5564 RLINE  $(0,3*S)-(0,-6*S),0$ 
5568 RLINE  $-(US*S*3,-10*S),9$ 
5572 USING U$
5576 LPRINT X
5580 GOTO "F-7"
5584 REM -----
5588 "F-6"
5592 REM -----
5596 LPRINT "M"; $U$ ;"", $0$ "
5600 LPRINT "R0,"; $3*S$ :LPRINT "J0,"; $-6*S$ 
5604 LPRINT "R"; $-US*S*3$ ;"", $-10*S$ 
5608 USING U$
5612 LPRINT "P";X
5616 REM -----
5620 "F-7"
5624 REM -----
5628 USING
5632 "KS-1" $X=X-XS$ : $U=X*XF$ :GOTO "KS-3"
5636 "KS-2"
```

Do not sale !

```
5640 REM -----
5644 REM ZEICHNE Y-ACHSE
5648 REM -----
5652 IF D$=""LINE (0,UK)-(0,UG),0
5656 IF D$<>""LPRINT "M0,";UK:LPRINT "D0,";UG
5660 Y=INT (YG/YS)*YS
5664 "KS-6"
5668 U=Y*YF
5672 IF U=0GOTO "KS-4"
5676 IF U<UKGOTO "KS-5"
5680 IF D$<>""GOTO "F-8"
5684 GLCURSOR (0,U)
5688 RLINE (3*S,0)-(-6*S,0),0
5692 RLINE -(-(US+1)*S*6,-3*S),9
5696 USING U$
5700 LPRINT Y
5704 GOTO "F-9"
5708 REM -----
5712 "F-8"
5716 REM -----
5720 LPRINT "M0,";U
5724 LPRINT "R";3*S;"",0":LPRINT "J";-6*S;"",0"
5728 LPRINT "R";-(US+1)*S*6;"",";-3*S
5732 USING U$
5736 LPRINT "P";Y
5740 REM -----
5744 "F-9"
5748 REM -----
5752 USING
5756 "KS-4"Y=Y-YS:GOTO "KS-6"
5760 "KS-5"
5764 IF D$=""LINE (UK,UK)-(UG,UK)-(UG,UG)-(UK,UG)-(UK,UK),0

5768 IF D$=""GLCURSOR (0,0)
5772 IF D$<>""LPRINT "M";UK;"",";UK
5776 IF D$<>""LPRINT "D";UG;"",";UK;"",";UG;"",";UG;"",";UK;"",
;UG
5780 IF D$<>""LPRINT "D";UK;"",";UK:LPRINT "H"
5784 RETURN
5788 REM -----
5792 REM ABFANGEN VON FEHLERN
5796 REM -----
5800 "ERROR"FE=PEEK &789B:IF FE>36AND FE<40RETURN
5804 WAIT :PRINT "ERROR ";FE
```

Do not sale !

In der ersten Zeile wird die BASIC-Formel für die zu zeichnende Funktion einprogrammiert. Bitte beachten Sie, daß die Nummer dieser Zeile in den Zeilen 5188, 5284 und 5456 zum Aufruf der Funktion benötigt wird. Nach dem Programmstart mit RUN "F" wird die Eingabe von minimalen und maximalen Werten für die x- und y-Achse verlangt, weiterhin die Schrittweite der Achsenbeschriftung und die Angabe der Bildgröße. Auch hier wird wieder das USING-Format zur Beschriftung der Achsen eingegeben.

Mit einem eigenen Unterprogramm "K-SYSTEM" wird das Koordinatensystem gezeichnet. (Das Zeichnen eines Koordinatensystems ist so grundlegend, daß dafür ein eigenes Unterprogramm angelegt wurde.) Zuerst werden die Umrechnungsfaktoren für x-Werte in u-Werte und y-Werte in v-Werte der Zeichnung bestimmt. Danach die größten und kleinsten u- und v-Werte (Zeile 5488 bis 5508). Es wird der Koordinatenursprung festgelegt und die x-Achse gezeichnet. Die Beschriftung der x-Achse beginnt beim größten möglichen Wert (Zeile 5540) und geht schrittweise zurück bis zum kleinsten. Dabei wird der Wert $x = 0$ ausgelassen. Das Zeichnen der Achsmarkierungen und die Beschriftung ist analog zum Balkendiagrammprogramm. Nach der x-Achse wird die y-Achse in gleicher Weise gezeichnet.

Nachdem das Koordinatensystem geplottet ist, erfolgt das Zeichnen der Funktion. Diese wird, beginnend bei dem kleinsten x-Wert, Punkt für Punkt berechnet und gezeichnet.

Es erhebt sich natürlich die Frage, wie dicht die zu berechnenden Punkte aufeinander liegen sollen. Man könnte, um ein möglichst genaues Bild zu erhalten, die Punkte im Abstand der Plotterauflösung berechnen: In Bildkoordinaten also mit dem Abstand von einer Einheit. Das erfordert aber eine hohe Rechenzeit. Eigentlich liegt es nahe, den Punktabstand dort, wo die Funktionswerte sich kaum ändern, groß und dort, wo die Funktionswerte sich sehr ändern, klein zu wählen. Diese "automatische Schrittweitenanpassung" wird in den Zeilen 5320 bis 5332 geregelt. Als Maß für die Änderung

Do not sale !

der Funktionswerte dient die Ableitung der Funktion

$$A = \frac{V_{\text{neu}} - V_{\text{alt}}}{U_{\text{neu}} - U_{\text{alt}}} = \frac{V_{\text{neu}} - V_{\text{alt}}}{\text{Schrittweite SW}}$$

Die Indizes neu bzw. alt beziehen sich auf den eben berechneten bzw. den zuvor berechneten Punkt. Im Programm ist die Schrittweitenanpassung so gewählt, daß die Schrittweite vergrößert wird, wenn sich alte und neue Ableitung um weniger als 10% unterscheiden:

$$\left| \frac{A_{\text{neu}} - A_{\text{alt}}}{A_{\text{alt}}} \right| < 10\% = 0.1$$

Wenn sich die neue und die alte Ableitung um mehr als 20% unterscheiden, dann wird die Schrittweite verkleinert:

$$\left| \frac{A_{\text{neu}} - A_{\text{alt}}}{A_{\text{alt}}} \right| > 20\% = 0.2$$

Die Vergrößerung der Schrittweite erfolgt durch Verdoppelung, die Verkleinerung der Schrittweite durch Halbierung der bisherigen. Dabei wird keine Schrittweite kleiner 1 oder größer 16 zugelassen.

Eine weitere Abfrage kontrolliert, ob der berechnete Funktionswert innerhalb des y-Wertebereiches liegt. Falls das nicht so ist, wird die Funktion nur bis zum Rand gezeichnet. Weiterhin wird dann die Schrittweite aus Sicherheitsgründen auf 1 reduziert. Dadurch wird der "Wiedereintritt" der Funktion in den darstellbaren Bildbereich nicht übersehen.

Eine letzte Kontrolle verhindert den Programmabbruch, wenn der Funktionswert an einer nicht definierten Stelle berechnet werden sollte. Z.B.

Do not sale !

$$y = \frac{1}{x} \quad \text{an der Stelle } x = 0$$

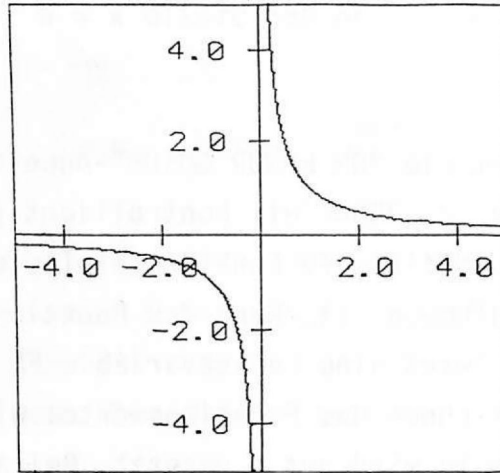
In diesem Fall verhindert die "ON ERROR GOSUB"-Anweisung den Programmabbruch. Es wird im ERROR-Teil kontrolliert (Zeile 5800), ob der Fehler auf eine undefinierte Funktionsstelle oder auf andere Ursachen zurückzuführen ist. Wenn der Funktionswert nicht berechnet werden konnte, wird eine Fehlervariable FE auf einen Wert $\neq 0$ gesetzt, das Zeichnen des Funktionswertes wird unterdrückt und die Schrittweite wird auf 1 gesetzt. Bei anderen Fehlerursachen bricht das Programm mit einer Fehlermeldung ab.

Achtung: Beim PC-1500 (A), für den die in diesem Buch beschriebenen Programme erstellt wurden, existiert kein BASIC-Kommando, das die letzte Fehlernummer angibt. Deswegen muß in der Zeile 5800 die Fehlernummer aus dem System-RAM gepeekt werden. Dies geht natürlich bei anderen BASIC-Rechnern auf andere Weise. Das Programm muß dementsprechend geändert werden. Gleiches gilt für die Fehlernummer: Beim PC-1500 (A) sind die Fehlernummern 37 ... 39 für nicht berechenbare Funktionswerte reserviert. Bei anderen Rechnern werden das andere Nummern sein. (Die CE-158-Schnittstelle enthält übrigens eine ERN-Funktion zur Bestimmung der Fehlernummer.)

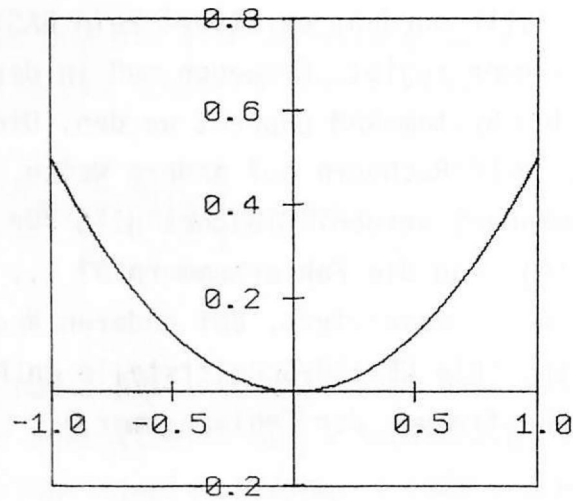
Nach dem Zeichnen der Funktion wird die Zeile 5000, die ja die BASIC-Formel der Funktion erhält, mit LLIST ausgedruckt.

Nachfolgend einige Funktionsbeispiele. Beachten Sie bitte beim Plotten, wie der Plotter immer wieder die Schrittweite an den Funktionsverlauf anpaßt.

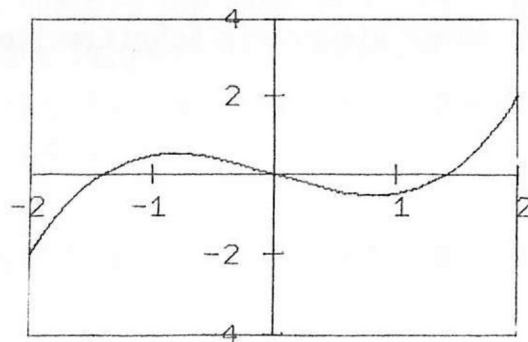
Do not sale !



5000 $Y=1/X$

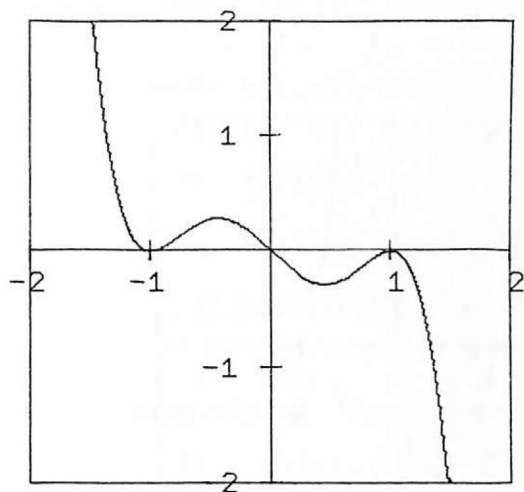


5000 $Y=.5*X*X$

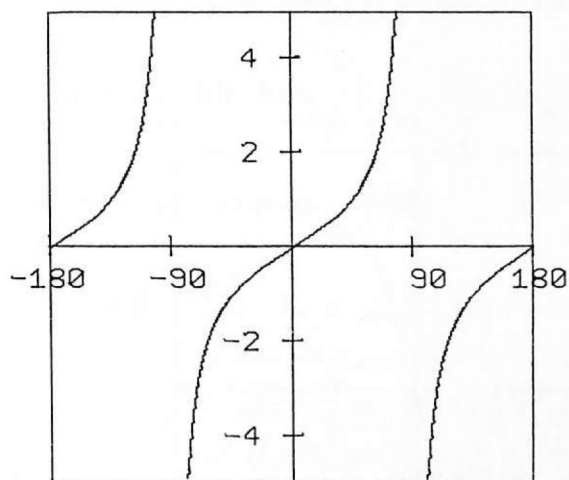


5000 $Y=.5*X^3-X$

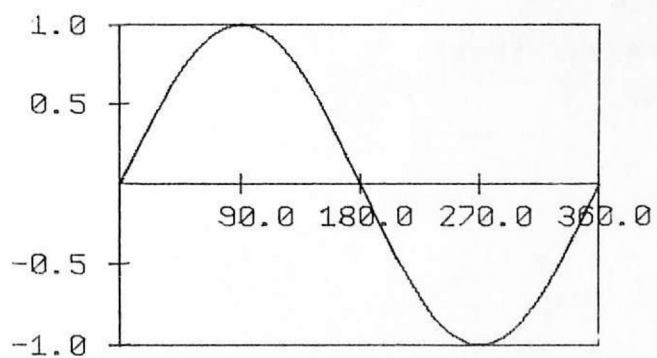
Do not sale !



5000 $Y = -x^5 + 2x^3 - x$

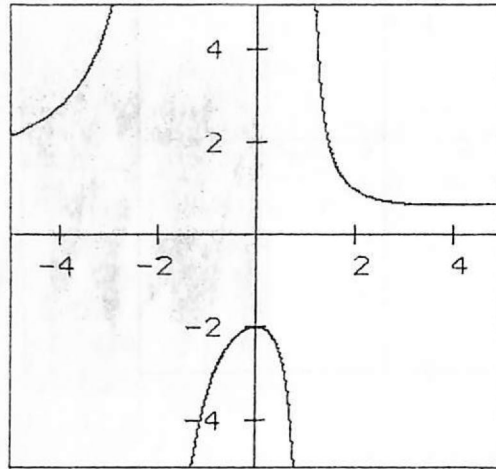


5000 $Y = \tan X$

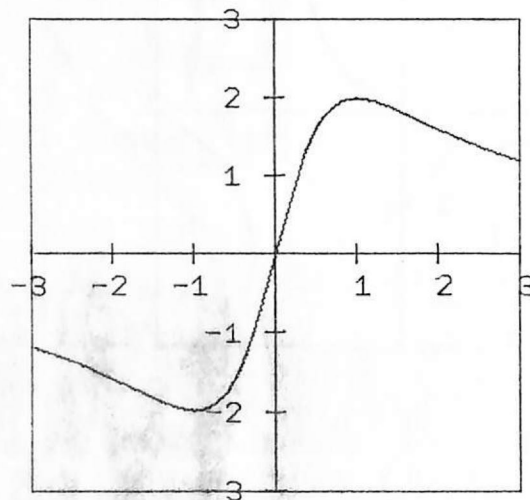


5000 $Y = \sin X$

Do not sale !



5000 $Y = (X^2 - 2X + 4) / (X^2 + X - 2)$



5000 $Y = 4X / (X^2 + 1)$

Do not sale !

PROGRAMM-LISTINGS FÜR PC-1500 (A):

- : Funktionen - Schaubilder mit dem PC-1500 und dem Farbdrucker CE-150. In: Zeitschrift HOBBY COMPUTER, Nr. 11/1983
- : Funktionsanalyse für PC-1500. In: Zeitschrift CHIP, Nr. 6/1983
(Programm berechnet auch Nullstellen, relative Extremwerte und Wendepunkte.)
- : Die Analyse mathematischer Funktionen mit dem PC-1500.
In: Zeitschrift PPX, Nr. 3/1983
(Programm berechnet auch relative Extremwerte, die Ableitung und das Integral der Funktion und die Umkehrfunktion.)

PROGRAMM-KASSETTEN FÜR PC-1500 (A):

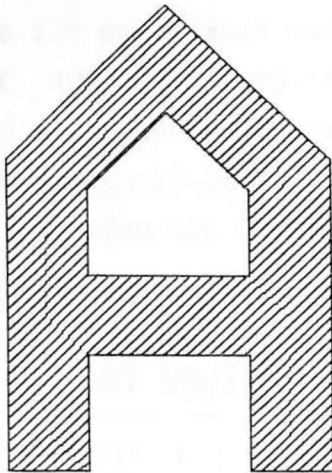
(Erhältlich bei der Fischel GmbH, Berlin)

- : FUNCTIONEN-PLOT (Komfortables Programm zum Zeichnen von Funktionen. Automatische Schrittweitenanpassung, kartesische und Polar-Koordinaten sind möglich. Es können mehrere Funktionen in ein Bild gezeichnet werden.)

Do not sale !

7 SCHRAFFUR VON VIELECKEN

Ein ganz anderes Problem als das Zeichnen von Bildern, Diagrammen und Funktionen ist das Schraffieren von Polygonen (Vielecken). Dabei dürfen diese Figuren auch Bereiche enthalten, die nicht schraffiert werden sollen (sogenannte Inseln):



Bevor der Schraffuralgorithmus vorgestellt wird, noch einige Hinweise zum Abspeichern solcher Figuren:

Zunächst werden alle Eckpunkte der Figur numeriert, und die Koordinaten dieser Punkte werden bestimmt. Danach wird ein Linienzug erstellt, der alle Punkte in geeigneter Weise miteinander verbindet. Um zu den Inseln einer Figur zu gelangen, werden speziell gekennzeichnete unsichtbare Linien eingeführt. Alle Bildangaben werden in DATA-Zeilen gespeichert.

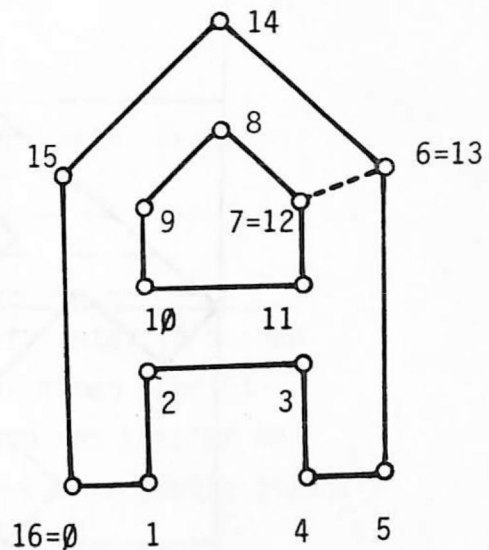
Die erste Zahl gibt an, aus wievielen Punkten die Figur besteht. Dann folgen DATA-Zeilen mit der ersten Zahl

0, wenn die Linie zum Punkt nicht gezeichnet werden soll

1, wenn die Linie zum Punkt gezeichnet werden soll.

Do not sale !

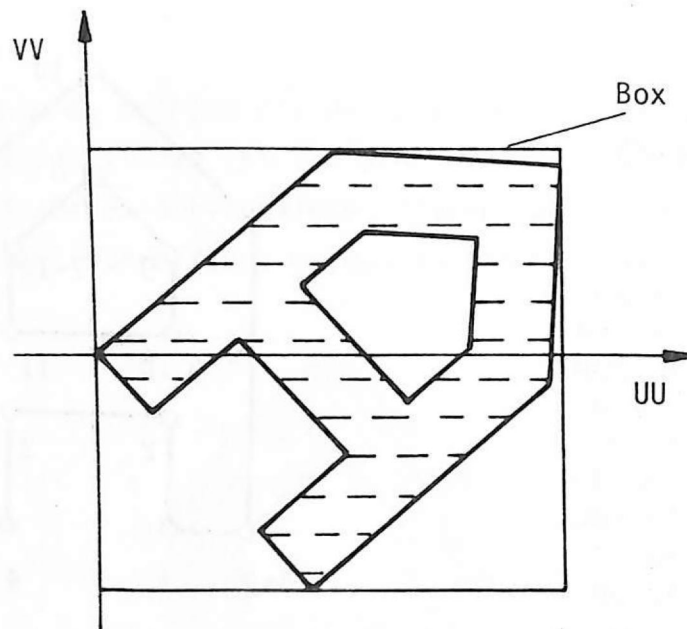
```
6730 "BUCHSTABE A"  
6735 DATA 16  
6740 DATA 1,0,0  
6745 DATA 1,50,0  
6750 DATA 1,50,75  
6755 DATA 1,150,75  
6760 DATA 1,150,0  
6765 DATA 1,200,0  
6770 DATA 1,200,200  
6775 DATA 0,150,180  
6780 DATA 1,100,230  
6785 DATA 1,50,180  
6790 DATA 1,50,125  
6795 DATA 1,150,125  
6800 DATA 1,150,180  
6805 DATA 0,200,200  
6810 DATA 1,100,300  
6815 DATA 1,0,200  
6820 DATA 1,0,0
```



Die zweite und dritte Zahl sind die U,V-Koordinaten des Punktes.
Der Anfangspunkt erscheint als letzter Punkt nochmal, um
das Vieleck zu schließen.

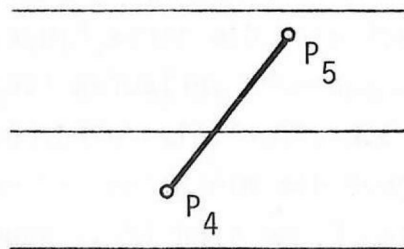
Der Algorithmus zum Berechnen der Schraffurlinien stammt, zumindest
teilweise, aus dem Buch von POMASKA und funktioniert folgendermaßen
(Die Zeilenangaben beziehen sich auf das Programm):

Zunächst wird die Schraffurrichtung, der Winkel der Schraffur zur
U-Achse, festgelegt. Danach wird die ganze Figur so transformiert,
daß die Schraffurlinien waagrecht verlaufen (vergl. Abs. 3.1.4).
Der Abstand zwischen den Schraffurlinien wird ebenfalls festgelegt.
Es wird untersucht, wie groß die minimalen und maximalen U- und
V-Werte der transformierten Figur sind (Berechnung der "Box",
in der das Vieleck liegt). (Zeile 6260 ... 6300)



Nun wird eine Schraffurlinie nach der anderen mit allen Kanten der Figur verglichen. Dabei werden eventuelle Schnittpunkte berechnet.

Wir betrachten zum Beispiel die Kante vom Punkt 4 zum Punkt 5 des Polygons. Zu den Punkten gehören die transformierten Koordinaten $UU(4)$, $VV(4)$ und $UU(5)$, $VV(5)$.



Die Berechnung des Schnittpunktes mit einer Schraffurlinie ist sehr einfach, da die Schraffurlinie durch die geschickte Transformation waagrecht liegt.

Do not sale !

Zunächst muß gelten, daß die VV-Koordinate der Schraffurlinie zwischen den Werten VV(4) und VV(5) liegt. Falls also

$$VV > VV(4) \quad \text{und} \quad VV > VV(5),$$

dann liegt die Schraffurlinie oberhalb der Kante. Falls

$$VV < VV(4) \quad \text{und} \quad VV < VV(5),$$

dann liegt die Schraffurlinie unterhalb der Kante. In beiden Fällen gibt es keinen Schnittpunkt. Wenn es einen Schnittpunkt gibt, dann kann man die UU-Koordinaten des Punktes berechnen mit der Geradengleichung der Körperkanten (Zeile 6505):

$$\frac{VV_S - VV(4)}{UU_S - UU(4)} = \frac{VV(5) - VV(4)}{UU(5) - UU(4)}$$

$$UU_S = UU(4) + \frac{VV_S - VV(4)}{VV(5) - VV(4)} * (UU(5) - UU(4))$$

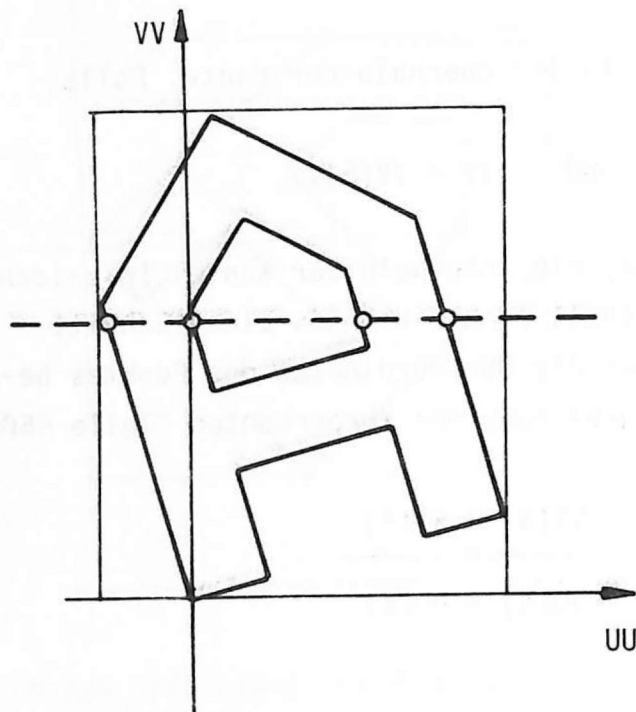
Die VV-Koordinate des Schnittpunktes ist natürlich VV_S .

Nachdem alle Kanten des Polygons mit einer Schraffurlinie zum Schnitt gebracht wurden, sortiert man die UU_S -Koordinaten aller Schnittpunkte in steigender Reihenfolge (Zeilen 6520 ... 6570). Man sieht (vergl. Bild auf der nächsten Seite), daß auf diese Art und Weise "Inseln" innerhalb des Polygons erkannt werden. Die Schraffurlinie wird

- vom 1. zum 2. Schnittpunkt gezeichnet
- vom 2. zum 3. Schnittpunkt nicht gezeichnet
- vom 3. zum 4. Schnittpunkt gezeichnet
- u.s.w.

Bevor aber die Schraffurlinienstücke geplottet werden können,

Do not sale !



müssen die UU- und VV-Koordinaten der Schnittpunkte rücktransformiert werden in das richtige Polygonkoordinatensystem U,V (Zeile 6575 ... 6605).

Auf die eben beschriebene Weise wird eine Schraffurlinie nach der anderen vom kleinsten VV-Wert bis zum größten mit allen Polygonkanten verglichen und gezeichnet.

Noch ein Hinweis: Der Algorithmus funktioniert nur dann, wenn die Schraffurlinien nicht exakt durch Eckpunkte des Polygons gehen. Deswegen wird vor dem Berechnen der Schnittpunkte einer Schraffurlinie mit dem Polygon der VV-Wert der Linie mit allen Punkt-VV-Werten verglichen (Zeile 6335 ... 6365). Falls (im Rahmen einer Rechengenauigkeit $\epsilon = EP = 0.000001$) die Linie durch einen Punkt geht, wird die Schraffurlinie einfach um den Wert EP nach oben verschoben. Diese geringfügige Verschiebung ist nicht sichtbar. Sie verhindert aber numerische

Do not sale !

Probleme im Algorithmus.

Und noch ein letzter Hinweis: Zu den Polygoninseln gelangt man über unsichtbare Linien. Der Schnitt der Schraffurlinie mit diesen unsichtbaren Linien wird unterdrückt (Zeile 6420). Im Feld IS(M) wird dazu für jede Linie eine Kenngröße gespeichert.

Nachfolgend das Programm und einige Beispiele:

```
6000 "S"
6005 REM -----
6010 REM  SCHRAFFUR EINES POLYGONS
6015 REM -----
6020 WAIT 0:DEGREE
6025 INPUT "Externer Drucker ? (J/N) ";D$
6030 IF D$<>"J"LET D$=""
6035 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PO
6040 IF D$<>" "CONSOLE 0,0
6045 IF D$=""GRAPH :GLCURSOR (0,-300):SORGN
6050 IF D$<>" "LPRINT CHR$(27)+"b":LPRINT "M0,-300"
6055 IF D$<>" "LPRINT "I"
6060 EP=.000001
6065 REM -----
6070 REM  LESEN DER POLYGON-DATEN
6075 REM  UND ZEICHNEN DES POLYGONS
6080 REM -----
6085 INPUT "Name der Figur ? ";N$
6090 RESTORE N$
6095 READ M
6100 DIM U(M),V(M),IS(M),UU(M),VV(M),US(10),VS(10)
6105 READ J,U,V
6110 IF D$=""GLCURSOR (U,V)
6115 IF D$<>" "LPRINT "M";U;" ";U
6120 U(0)=U
6125 V(0)=V
6130 IS(0)=J
6135 FOR I=1TO M
6140 READ J,U,V
6145 IF J=0AND D$=""LINE -(U,V),9
6150 IF J=0AND D$<>" "LPRINT "M";U;" ";U
6155 IF J=1AND D$=""LINE -(U,V),0
6160 IF J=1AND D$<>" "LPRINT "D";U;" ";U
6165 U(I)=U
6170 V(I)=V
6175 IS(I)=J
6180 NEXT I
```

Do not sale !

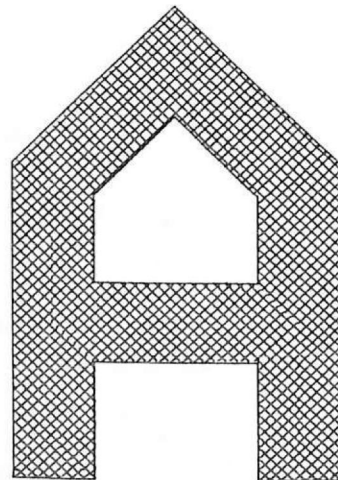
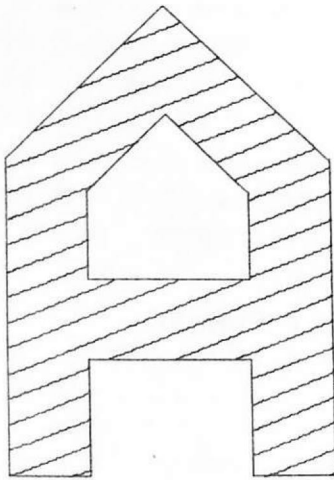

```
6185 REM -----
6190 "SCHRAFFUR"
6195 REM -----
6200 REM  EINGABE DER SCHRAFFURDATEN
6205 REM -----
6210 AL=45:INPUT "Winkel der Schraffur ? ";AL
6215 SS=10:INPUT "Schrittweite ? ";SS
6220 REM -----
6225 REM  TRANSFORMATION DES POLYGONS
6230 REM -----
6235 PRINT "Ich transformiere"
6240 FOR I=0 TO M
6245 UU(I)=(U(I)-U(0))*COS AL+(V(I)-V(0))*SIN AL
6250 VV(I)=-(U(I)-U(0))*SIN AL+(V(I)-V(0))*COS AL
6255 NEXT I
6260 REM -----
6265 REM  MIN. UND MAX. U-WERTE DES POLYGONS
6270 REM -----
6275 UK=UU(0)
6280 UG=UK
6285 FOR I=1 TO M-1
6290 IF UU(I)<UKLET UK=UU(I)
6295 IF VV(I)>UGLET UG=VV(I)
6300 NEXT I
6305 REM -----
6310 REM  SCHRAFFUR AB U=UK+.1*SS BIS U=UG
6315 REM -----
6320 U=UK+.1*SS
6325 "S-ANFANG"
6330 PRINT "U ; ";U
6335 REM -----
6340 REM  KEIN PUNKT DES POLYGONS DARF AUF DER AKTUELLEN
6345 REM  SCHRAFFURLINIE LIEGEN
6350 REM -----
6355 FOR I=0 TO M-1
6360 IF ABS (UU(I)-U)<ELET U=U+EP:GOTO "S-ANFANG"
6365 NEXT I
6370 REM -----
6375 REM  TEST ALLER POLYGONLINIEN AUF SCHNITT
6380 REM  MIT DER SCHRAFFURLINIE
6385 REM -----
6390 US=-1
6395 FOR I=0 TO M-1
6400 J=I+1
```

Do not sale !

```
6405 REM -----
6410 REM KEIN TEST, FALLS LINIE UNSICHTBAR
6415 REM -----
6420 IF IS(J)=0GOTO "S-Next I"
6425 REM -----
6430 REM BEST. DER LINIE
6435 REM -----
6440 U1=UU(I)
6445 U2=UU(J)
6450 V1=VV(I)
6455 V2=VV(J)
6460 REM -----
6465 REM BOX-TEST
6470 REM -----
6475 IF V1>VAND V2>VGOTO "S-Next I"
6480 IF V1<VAND V2<VGOTO "S-Next I"
6485 REM -----
6490 REM BEST. DES SCHNITTPUNKTES
6495 REM -----
6500 US=US+1
6505 US(US)=U1+(V-V1)/(V2-V1)*(U2-U1)
6510 "S-Next I"NEXT I
6515 IF US<1GOTO "NAECHSTE LINIE"
6520 REM -----
6525 REM SORTIEREN DER SCHNITTPUNKTE
6530 REM -----
6535 FOR I=0TO US-1
6540 SK=US(I)
6545 SJ=I
6550 FOR J=I+1TO US
6555 IF US(J)<SKLET SK=US(J):S.J=J
6560 NEXT J
6565 IF I<>SJLET US(SJ)=US(I):US(I)=SK
6570 NEXT I
6575 REM -----
6580 REM RUECKTRANSFORMATION DER SCHNITTPUNKTE
6585 REM -----
6590 FOR I=0TO US
6595 US(I)=U(0)+US(I)*SIN AL+V*COS AL
6600 US(I)=U(0)+US(I)*COS AL-V*SIN AL
6605 NEXT I
6610 REM -----
6615 REM ZEICHNEN DER SCHRAFFURLINIE
6620 REM -----
```

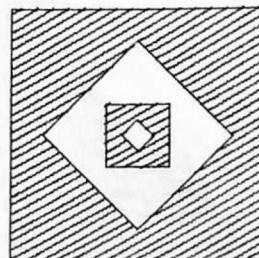
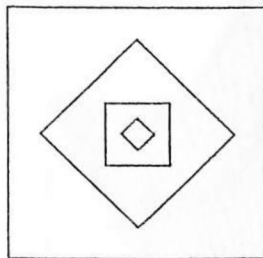
Do not sale !

```
6625 FOR I=1TO USSTEP 2
6630 IF D$=""LINE (US(I-1),VS(I-1))-(US(I),VS(I)),0
6635 IF D$<>""LPRINT "M";INT US(I-1);",,";INT US(I-1)
6640 IF D$<>""LPRINT "D";INT US(I);",,";INT US(I)
6645 NEXT I
6650 REM -----
6655 REM ENDE DER SCHLEIFE UEBER ALLE SCHRAFFURLINIEN
6660 REM -----
6665 "NAECHSTE LINIE"
6670 U=U+SS
6675 IF U<UGGOTO "S-ANFANG"
6680 REM -----
6685 REM WEITERE SCHRAFFUR ?
6690 REM -----
6695 IF D$=""GLCURSOR (0,0)
6700 IF D$<>""LPRINT "H"
6705 S$="";INPUT "Weitere Schraffur ? (J/N)";S$
6710 IF S$="J"GOTO "SCHRAFFUR"
6715 IF D$=""TEXT
6720 IF D$<>""LPRINT "A";SETDEV
6725 END
```



Do not sale !

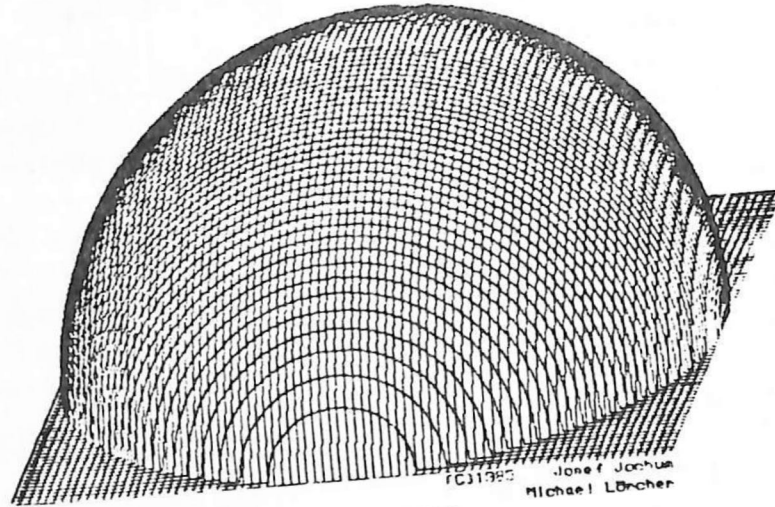
6825 "QUADRATE"
6830 DATA 20
6835 DATA 1,0,0
6840 DATA 1,160,0
6845 DATA 0,90,80
6850 DATA 1,80,90
6855 DATA 1,70,80
6860 DATA 1,80,70
6865 DATA 1,90,80
6870 DATA 0,100,60
6875 DATA 1,100,100
6880 DATA 1,60,100
6885 DATA 1,60,60
6890 DATA 1,100,60
6895 DATA 0,80,20
6900 DATA 1,140,80
6905 DATA 1,80,140
6910 DATA 1,20,80
6915 DATA 1,80,20
6920 DATA 0,160,0
6925 DATA 1,160,160
6930 DATA 1,0,160
6935 DATA 1,0,0



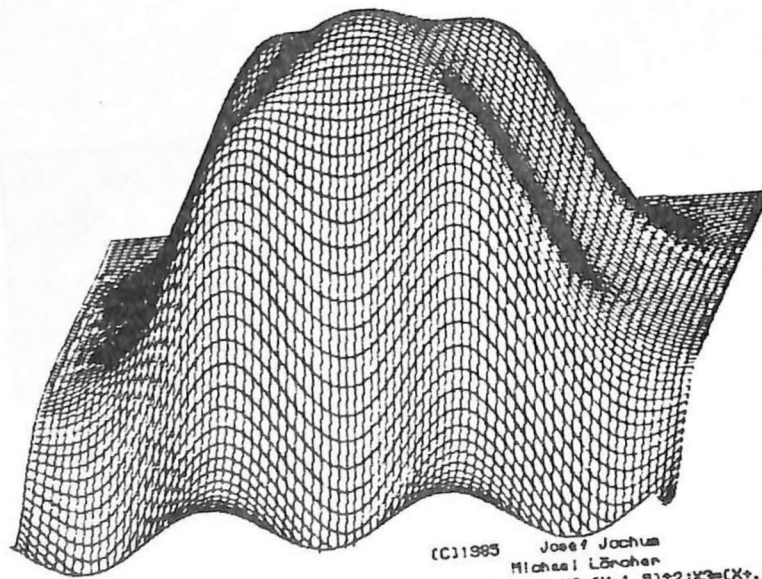
Do not sale !

LITERATUR:

: POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung.
1. Aufl.; Würzburg 1984



```
(C)1985 Josef Jochum  
Michael Lörcher  
030 R2=X*X+Y*Y:IF R2 > 4 THEN Z(I)=0:GOTO1000  
040 Z(I)=SQRT(4-R2)  
050 REM  
070 REM *****  
080 REM  
090 REM  
*** : X-Achse von -2 bis 2  
*** : Y-Achse von -2 bis 2  
Streckfaktor in Z-Richtung 1  
Aufbauung : .85  
Aufbauung : .85
```



```
(C)1985 Josef Jochum  
Michael Lörcher  
530 Y1=(Y+1.5)*2:Y2=(Y+.8)*2:Y3=(Y-.8)*2 :X1=(X+1.5)*2:X2=(X-1.5)*2:X3=(X+.8)*2  
2:XA=(X-.8)*2  
500 Z(I)=EXP(-1*(X*X+Y*Y))*2-EXP(3 *(-X1-Y1))-EXP(3 *(-X2-Y1))-EXP(3 *(-X3-Y1))  
+EXP(3*(-X3-Y2))+EXP(3 *(-X4-Y2))+EXP(3 *(-X3-Y3))+EXP(3 *(-X4-Y3))  
1000 REM  
*** : X-Achse von -2 bis 2  
*** : Y-Achse von -2 bis 2  
Streckfaktor in Z-Richtung 1  
Aufbauung : .85  
Aufbauung : .85
```

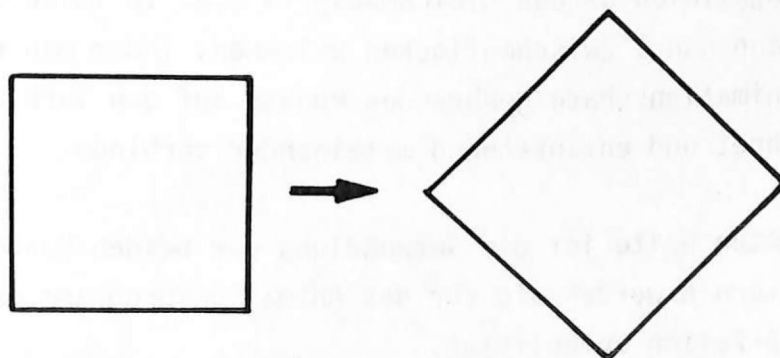
Do not sale !

8 COMPUTERANIMATION

Die Computeranimation ist ein wichtiges Teilgebiet der Computergrafik. Sie wird beim Erstellen von Zeichentrickfilmen und bei der Darstellung sich bewegender Figuren auf Bildschirmen verwendet. Diese beiden Aufgaben übersteigen eigentlich die Möglichkeiten der hier beschriebenen Plotter. Telespiele und anspruchsvolle Bildschirmgrafiken erfordern eben einen Video-Anschluß und keinen Plotter. Und wer einen Zeichentrickfilm erstellen will, wird wohl auf professionellere (und teurere) Geräte zurückgreifen als die hier besprochenen. Wenn wir trotzdem das Gebiet der Computeranimation streifen, dann nur um die prinzipielle Vorgehensweise zu erläutern.

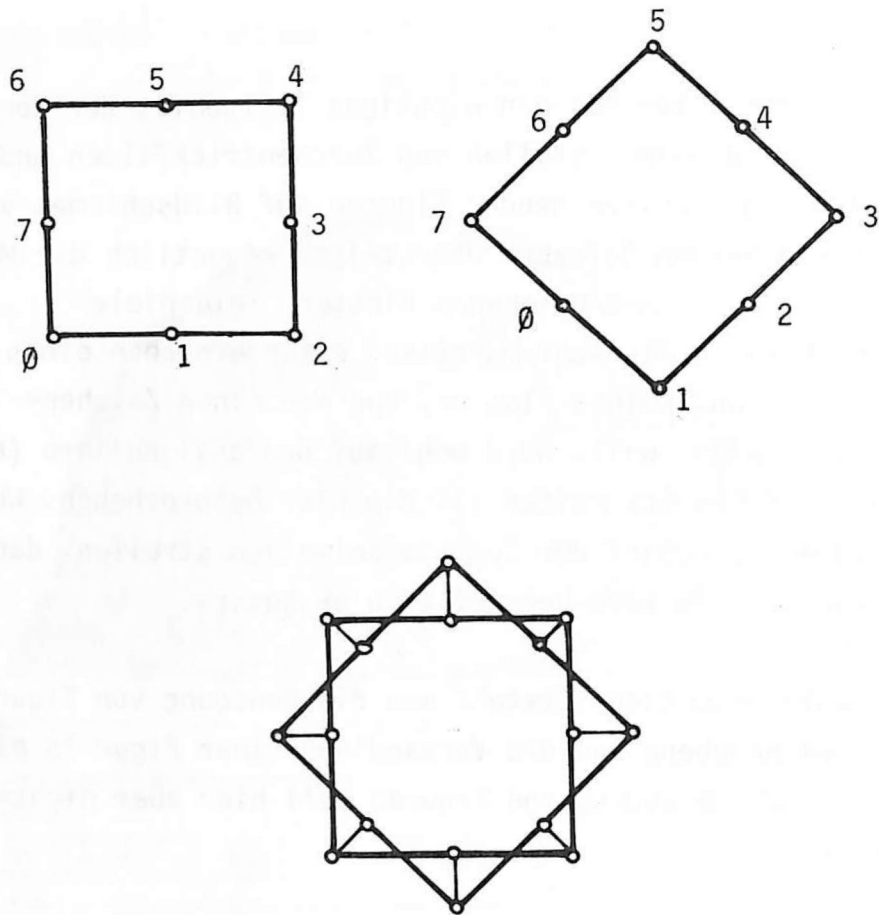
Unter Computeranimation versteht man die Bewegung von Figuren über die Zeichenebene und die Verwandlung einer Figur in eine andere. Über die Bewegung von Figuren soll hier aber nicht gesprochen werden.

Wir wollen ein einfaches Beispiel einer Verwandlung betrachten: Ein Quadrat soll in ein anderes, gedrehtes Quadrat verwandelt werden:



Man ordnet nun ausgewählte Punkte beider Figuren einander zu und verbindet sie (in Gedanken) mit Linien:

Do not sale !

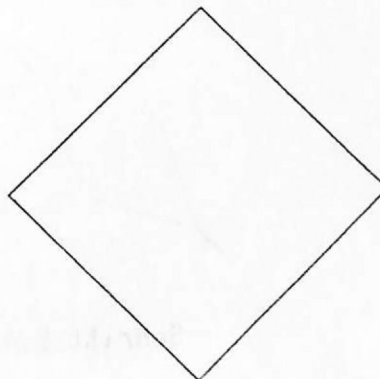
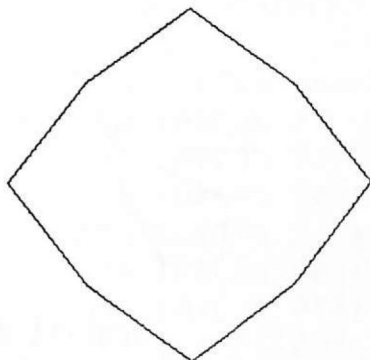
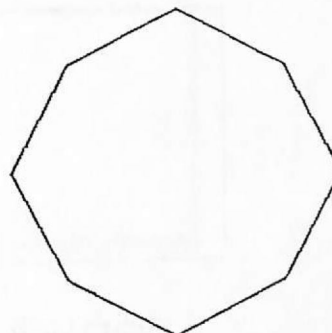
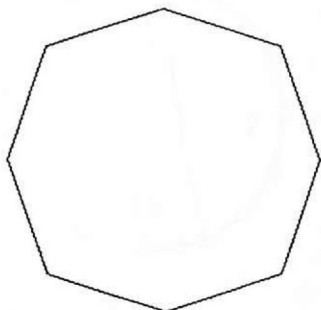
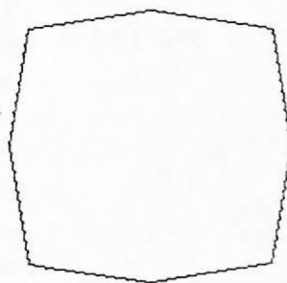
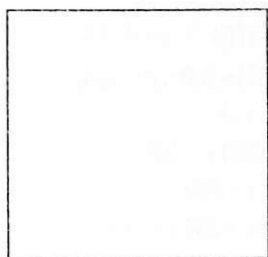


Die Verbindungslinien werden gleichmäßig in z.B. 10 Teile unterteilt. Man kann nun 9 Zwischenfiguren zeichnen, indem man die zur jeweiligen Animationsphase gehörenden Punkte auf den Verbindungslinien berechnet und entsprechend miteinander verbindet.

Auf der nächsten Seite ist die Verwandlung der beiden Quadrate gezeigt. Weiterhin werden die für das Animationsprogramm benötigten DATA-Zeilen aufgelistet.

Richtigen Trickfilmfreunden ist diese Animation aber zu "brav". Der Übergang vom einen Quadrat zum anderen ist nicht ausgeprägt genug. Dem kann man abhelfen, indem man andere Punkte einander

Do not sale !

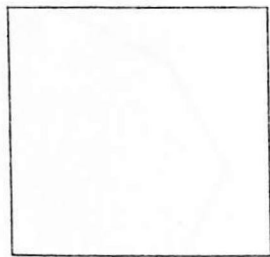


```
7592 "QUADRAT"  
7594 DATA 8  
7596 DATA -40,-40,-30,-30  
7598 DATA 0,-40,0,-60  
7600 DATA 40,-40,30,-30  
7602 DATA 40,0,60,0  
7604 DATA 40,40,30,30  
7606 DATA 0,40,0,60  
7608 DATA -40,40,-30,30  
7610 DATA -40,0,-60,0  
7612 DATA -40,-40,-30,-30  
7614 DATA -999
```

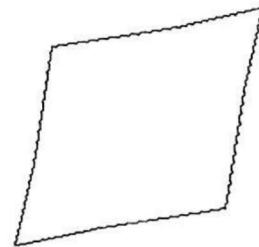
Do not sale !

zuordnet:

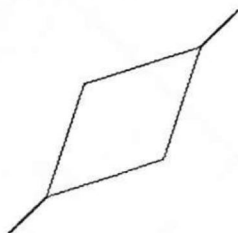
```
7640 "QUADRAT-NEU"  
7642 DATA 8  
7644 DATA -40,-40,-30,-30  
7646 DATA 0,-40,-60,0  
7648 DATA 40,-40,-30,30  
7650 DATA 40,0,0,60  
7652 DATA 40,40,30,30  
7654 DATA 0,40,60,0  
7656 DATA -40,40,30,-30  
7658 DATA -40,0,0,-60  
7660 DATA -40,-40,-30,-30  
7662 DATA -999
```



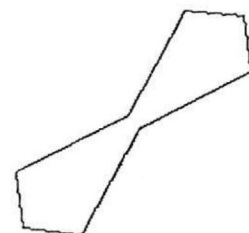
Schritt 0



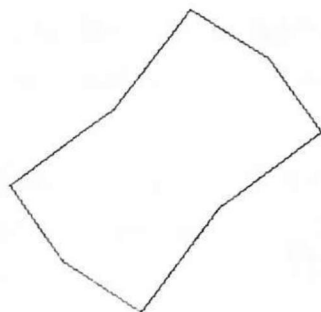
Schritt 0.2



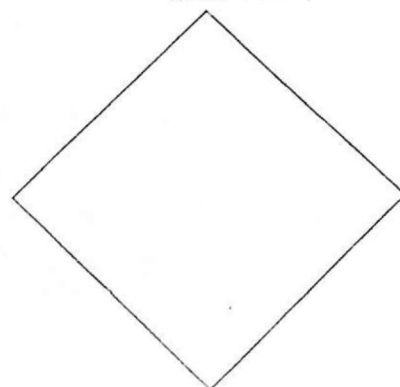
Schritt 0.4



Schritt 0.6



Schritt 0.8

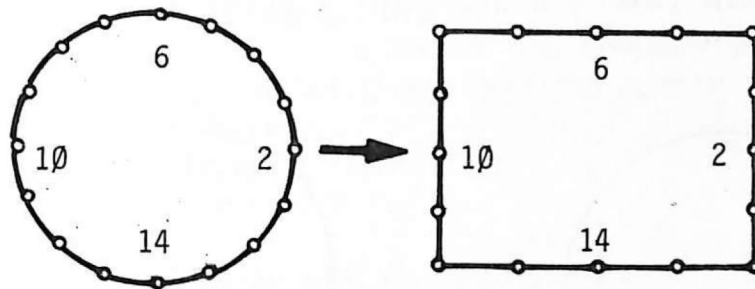


Schritt 1

Jetzt sieht das Ganze schon professioneller aus.

Do not sale !

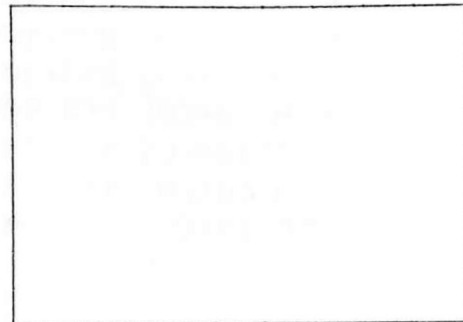
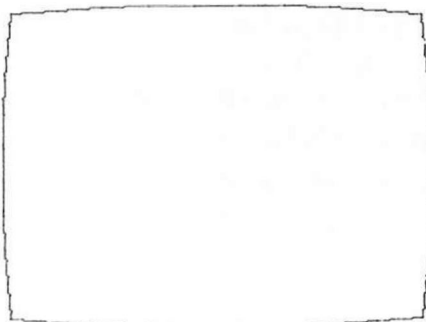
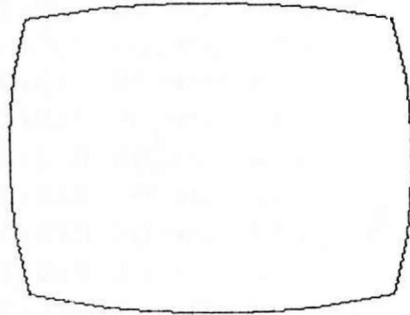
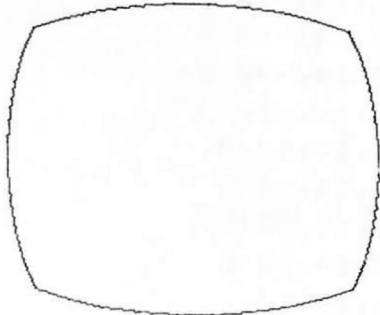
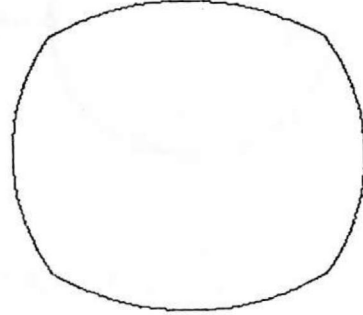
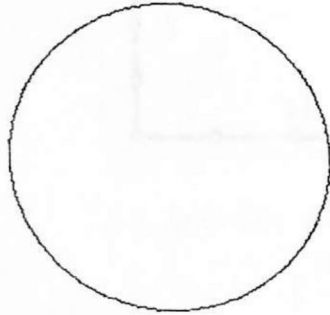
Bisher wurden nur gerade Linien in gerade Linien übergeführt.
Man kann natürlich auch einen Kreis in ein Rechteck verwandeln.
Dazu wird der Kreis als Vieleck, z.B. als 16-Eck aufgefaßt, wie
wir das schon im Abschnitt 3.1.9 beschrieben haben. Man be-
rechnet dann einzelne Punkte der "Kreises" (16-Ecks) und ordnet
sie entsprechenden Punkten des Rechtecks zu:



```
7512 "KREIS-RECHTECK"  
7514 DATA 36  
7516 DATA 10*COS 315,10*SIN 315,14,-10  
7518 DATA 10*COS 325,10*SIN 325,14,-70/9  
7520 DATA 10*COS 335,10*SIN 335,14,-50/9  
7522 DATA 10*COS 345,10*SIN 345,14,-30/9  
7524 DATA 10*COS 355,10*SIN 355,14,-10/9  
7526 DATA 10*COS 005,10*SIN 005,14,10/9  
7528 DATA 10*COS 015,10*SIN 015,14,30/9  
7530 DATA 10*COS 025,10*SIN 025,14,50/9  
7532 DATA 10*COS 035,10*SIN 035,14,70/9  
7534 DATA 10*COS 045,10*SIN 045,14,10  
7536 DATA 10*COS 055,10*SIN 055,98/9,10  
7538 DATA 10*COS 065,10*SIN 065,70/9,10  
7540 DATA 10*COS 075,10*SIN 075,42/9,10  
7542 DATA 10*COS 085,10*SIN 085,14/9,10  
7544 DATA 10*COS 095,10*SIN 095,-14/9,10  
7546 DATA 10*COS 105,10*SIN 105,-42/9,10  
7548 DATA 10*COS 115,10*SIN 115,-70/9,10  
7550 DATA 10*COS 125,10*SIN 125,-98/9,10  
7552 DATA 10*COS 135,10*SIN 135,-14,10  
7554 DATA 10*COS 145,10*SIN 145,-14,70/9  
7556 DATA 10*COS 155,10*SIN 155,-14,50/9
```

Do not sale !

7558 DATA 10*COS 165,10*SIN 165,-14,30/9
7560 DATA 10*COS 175,10*SIN 175,-14,10/9
7562 DATA 10*COS 185,10*SIN 185,-14,-10/9
7564 DATA 10*COS 195,10*SIN 195,-14,-30/9
7566 DATA 10*COS 205,10*SIN 205,-14,-50/9
7568 DATA 10*COS 215,10*SIN 215,-14,-70/9
7570 DATA 10*COS 225,10*SIN 225,-14,-10
7572 DATA 10*COS 235,10*SIN 235,-98/9,-10
7574 DATA 10*COS 245,10*SIN 245,-70/9,-10
7576 DATA 10*COS 255,10*SIN 255,-42/9,-10
7578 DATA 10*COS 265,10*SIN 265,-14/9,-10
7580 DATA 10*COS 275,10*SIN 275,14/9,-10
7582 DATA 10*COS 285,10*SIN 285,42/9,-10
7584 DATA 10*COS 295,10*SIN 295,70/9,-10
7586 DATA 10*COS 305,10*SIN 305,98/9,-10
7588 DATA 10*COS 315,10*SIN 315,14,-10
7590 DATA -999



Do not sale !

Das Programm, mit dem diese Bilder erstellt wurden, ist nachfolgend beschrieben:

```
7000 "A"
7005 REM -----
7010 REM  COMPUTER-ANIMATION
7015 REM -----
7020 WAIT 0:DEGREE
7025 INPUT "Externer Drucker ? (J/N) ";D$
7030 IF D$<>"J"LET D$=""
7035 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PD
7040 IF D$<>" "CONSOLE 0,0
7045 IF D$=""GRAPH :GLCURSOR (100,-100):SORGN
7050 IF D$<>" "LPRINT CHR$ (27)+"b":LPRINT "M200,-200"
7055 IF D$<>" "LPRINT "I"
7060 INPUT "Name ? ";N$
7065 INPUT "Faktor ? ";F
7070 REM -----
7075 REM  EINGABE DER ANIMATIONSPHASE
7080 REM -----
7085 "A-Schritt"INPUT "Schritt (0 ... 1) ? ";S
7090 IF D$=""AND (S<0OR S>1)TEXT :END
7095 IF D$<>" "AND (S<0OR S>1)LPRINT "A":SETDEV :END
7100 INPUT "COLOR ? ";C
7105 IF D$=""COLOR C
7110 IF D$<>" "LPRINT CHR$ (27)+CHR$ (&30+C)
7115 A$="N":INPUT "Nullpkt verschieben ? ";A$
7120 U=0:IF A$="J"INPUT "Wie weit ? ";U
7125 IF D$=""GLCURSOR (0,-ABS U):SORGN
7130 IF D$<>" "LPRINT "M0,";-ABS U:LPRINT "I"
7135 WAIT 0:PRINT "Bitte Geduld! Ich arbeite!"
7140 L=1:IF S=0OR S=1LET L=0
7145 IF D$<>" "LPRINT "L";L
7150 RESTORE N$
7155 REM -----
7160 REM  NEUE TEILFIGUR
7165 REM -----
7170 "A-Weiter"
7175 READ M
7180 IF M=-999GOTO "A-Ende"
```

Do not sale !

```
7185 REM -----
7190 REM  SCHLEIFE UEBER ALLE PUNKTE
7195 REM -----
7200 FOR K=0 TO M
7205 REM -----
7210 REM  LESEN UND BERECHNEN DER PUNKTE
7215 REM -----
7220 READ U0,U0,U1,U1
7225 U=(U0+S*(U1-U0))*F
7230 U=(U0+S*(U1-U0))*F
7235 REM -----
7240 REM  ZEICHNEN
7245 REM -----
7250 IF K=0 AND D$=""GLCURSOR (U,U)
7255 IF K=0 AND D$<>""LPRINT "M";INT U;",";INT U
7260 IF K<>0 AND D$=""LINE -(U,U),L
7265 IF K<>0 AND D$<>""LPRINT "D";INT U;",";INT U
7270 NEXT K
7275 GOTO "A-Weiter"
7280 "A-Ende"
7285 IF D$=""GLCURSOR (0,0)
7290 IF D$<>""LPRINT "H"
7295 GOTO "A-Schritt"
```

N\$ ist der Name der DATA-Zeilen, die die Koordinaten der Figurenpunkte enthalten. Der Faktor F ist ein Vergrößerungsfaktor, mit dem alle Koordinaten multipliziert werden können, um die ganze Zeichenebene auszufüllen. Mit Schritt (Zahl zwischen 0 und 1) ist die Animationsphase, die gezeichnet werden soll, gemeint.

Schritt = 0 bedeutet Ausgangsfigur

Schritt = 1 bedeutet Endfigur.

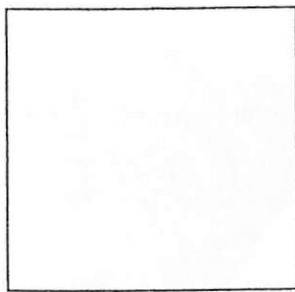
Mit der Nullpunktverschiebung kann man wählen, ob verschiedene Animationsphasen ineinander (Zwischenphasen werden gestrichelt) oder separat geplottet werden sollen. (Bei den hier gezeigten Zwischenphasen wurde die Strichelung der Linien unterdrückt, um eine bessere Druckqualität zu erhalten.) M gibt an, aus wievielen Teillinien ein Linienzug bestehen soll.

Do not sale !

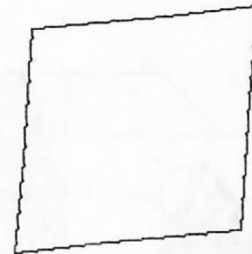
In den DATA-Zeilen sind dann die einzelnen Punktkoordinaten in der Form U_0, V_0, U_1, V_1 gespeichert. (U_0, V_0 : Koordinaten der einen Figur, U_1, V_1 : Zugeordnete Koordinaten der anderen Figur.) Die erste Zahl M gibt die Zahl der Punkte des ersten Linienzuges an. Der nullte Punkt wird mit gehobenem Stift angesteuert, die restlichen M Punkte werden mit gesenktem Stift durch Linien verbunden. Wenn dann die nächste DATA-Zeile ein -999 enthält, ist die Figur beendet. Ansonsten wird ein weiterer Linienzug gezeichnet.

Die Vorgabe einer Phase <0 oder >1 beendet den Animationslauf.

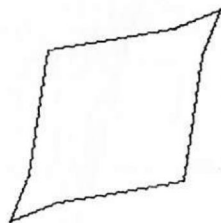
Nachfolgend sind einige weitere Umwandlungen mit den zugehörigen DATA-Zeilen aufgeführt.



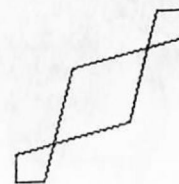
Schritt 0



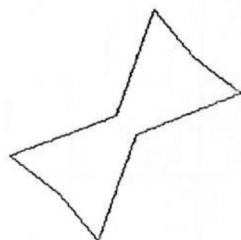
Schritt 0.2



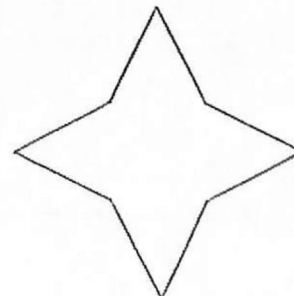
Schritt 0.4



Schritt 0.6



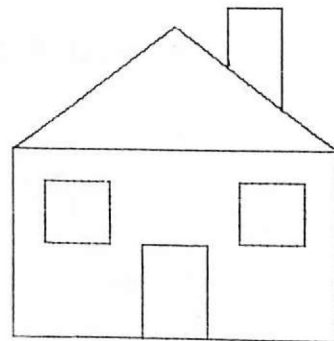
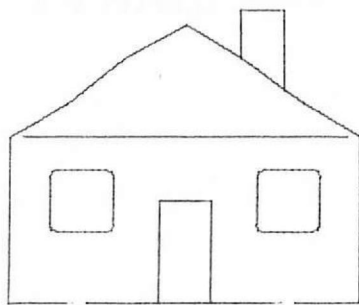
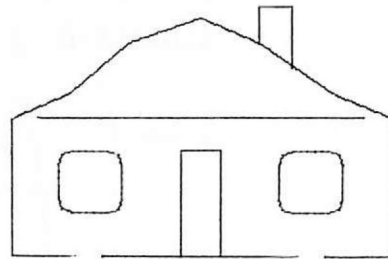
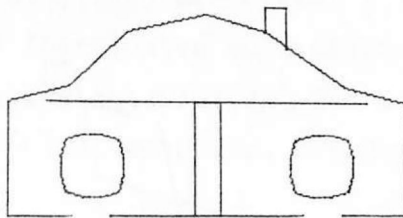
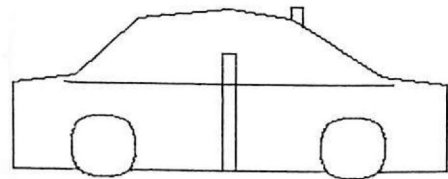
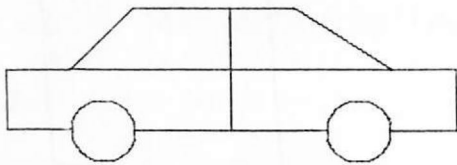
Schritt 0.8



Schritt 1

Do not sale !

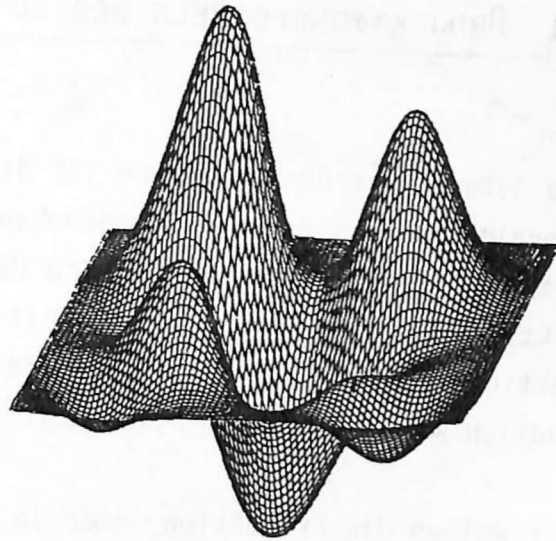
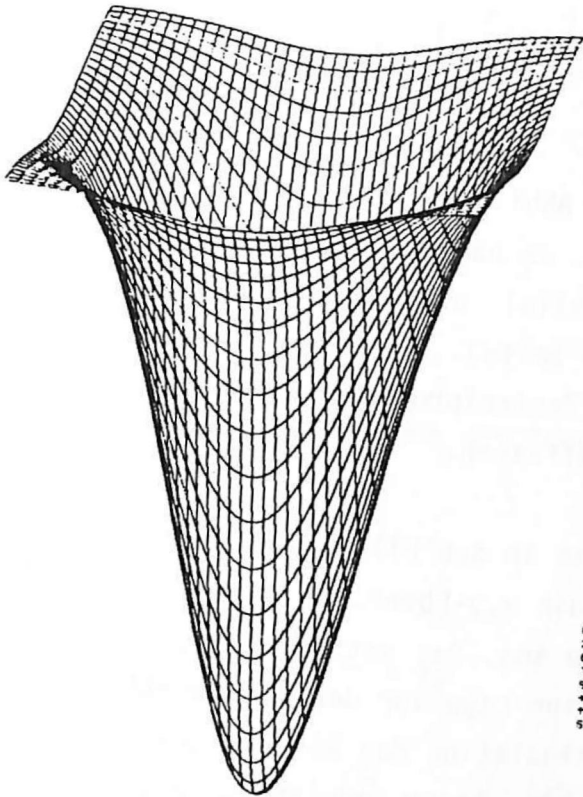
7616 "STERN"
7618 DATA 8
7620 DATA 0,90,90,0
7622 DATA 90,90,30,30
7624 DATA 90,0,0,90
7626 DATA 90,-90,-30,30
7628 DATA 0,-90,-90,0
7630 DATA -90,-90,-30,-30
7632 DATA -90,0,0,-90
7634 DATA -90,90,30,-30
7636 DATA 0,90,90,0
7638 DATA -999



Do not sale !

LITERATUR:

- : WADSWORTH, Nat: Introduction to Computer Animation.
New Jersey 1979
- : ANGELL, Ian O.: A Practical Introduction to Computer
Graphics. London 1981
- : POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung.
1. Aufl.; Würzburg 1984



```
FC1003 Josef Jochim  
Michael Lücher  
030 Y1=(Y1.1)Y2=(Y+.0)Y3=(Y-.0)Y2 X1=(X+1.1)Y2 X2=(X-1.1)Y2 X3=(X+.0)Y  
2 X4=(X-.0)Y2  
040 Z(1)=EXP(-18(X0+Y0))(-2)-EXP(30(-X2-Y1))+EXP(30(-X3-Y2))+EXP(30(-X4-Y3))  
1+EXP(30(-X3-Y3))+EXP(30(-X4-Y3))2.1-EXP(30(-X1-Y1))  
900 REF  
*** I X-Achse von -2 bis 2 Auflösung 1 .03  
*** I Y-Achse von -2 bis 2 Auflösung 1 .03  
Streckfaktor in Z-Richtung 1
```


9 GRUNDLAGEN DER 3D-GRAFIK

Alle bisher im Buch besprochenen Programme sind sogenannte 2D-Anwendungen, Darstellungen von zweidimensionalen Figuren. Ein anderes wichtiges Teilgebiet der Computergrafik ist die 3D-Grafik, die zweidimensionale Darstellung von dreidimensionalen Körpern auf der Zeichenebene.

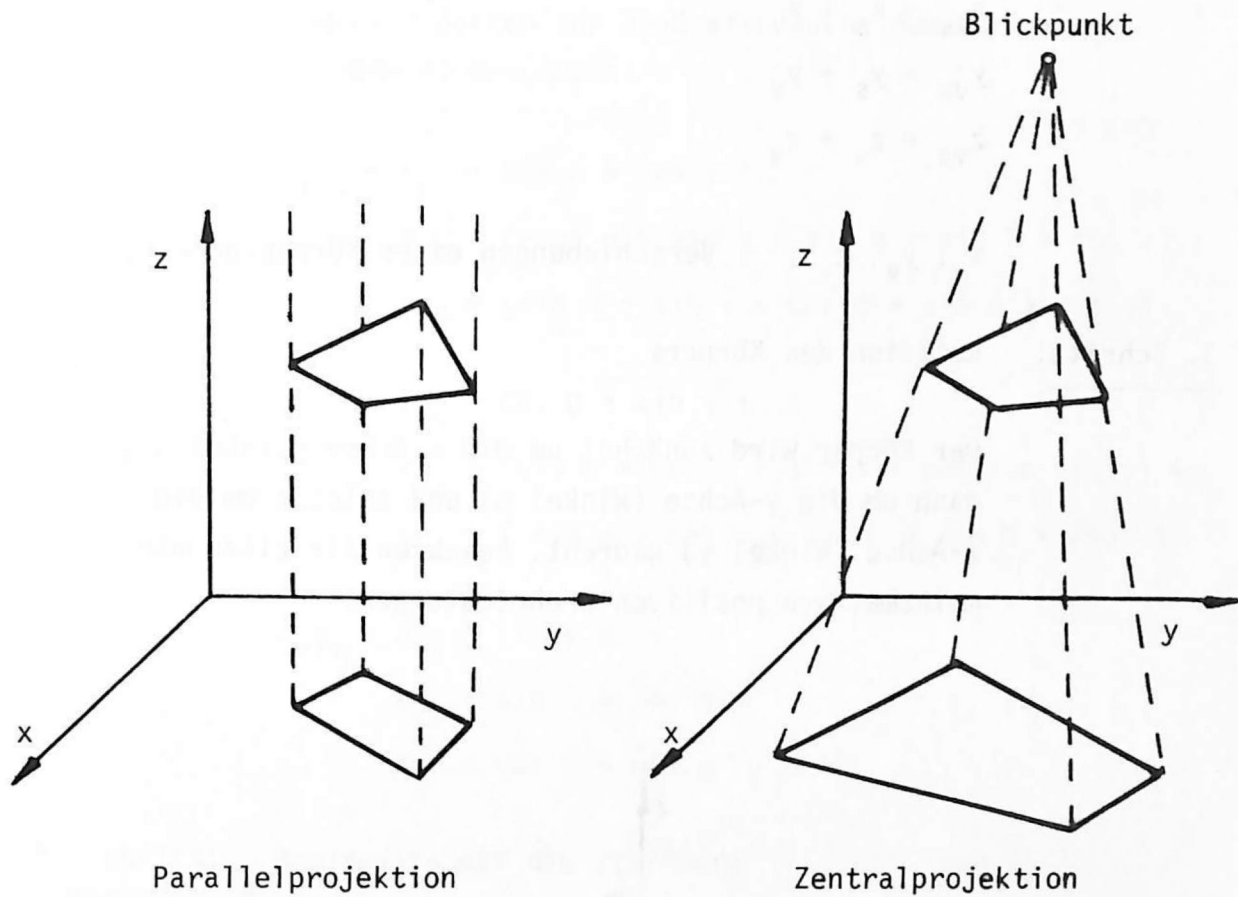
Die 3D-Grafik verlangt zunächst die Projektion des Körpers auf die Bildebene. Danach kann man das projizierte Bild auf die Unterdrückung nichtsichtbarer Linien hin untersuchen.

9.1 PROJEKTIONSFORMELN DER 3D-GRAFIK

Die Aufgabe der 3D-Projektion ist die Abbildung eines dreidimensionalen Körpers auf die Zeichenebene. Je nach Blickpunkt des Betrachters unterscheidet man die Parallel- und die Zentralprojektion (vergl. Bild auf der nächsten Seite). Die Parallelprojektion kann man als Spezialfall der Zentralprojektion mit unendlich weit entferntem Blickpunkt auffassen.

Wir wollen die Projektion immer so wie in den Bildern gezeigt durchführen: Nämlich Projektion auf die x,y -Ebene von einem Blickpunkt mit positiver z -Koordinate aus. Das setzt voraus, daß der Körper zunächst in die richtige Lage vor dem Blickpunkt verschoben und gedreht wird. Zur Manipulation des Bildes kann man die Körperabmessungen auch strecken, indem jede Koordinate mit einem Faktor vergrößert wird.

Do not sale !



1. Schritt: Skalierung der Körperkoordinaten

$$x_s = x * x_f$$

$$y_s = y * y_f$$

$$z_s = z * z_f$$

x, y, z : Koordinaten eines Körperpunktes

x_f, y_f, z_f : Skalierungs- (Vergrößerungs-) Faktoren

2. Schritt: Verschiebung des Körpers

$$x_{VS} = x_S + x_V$$

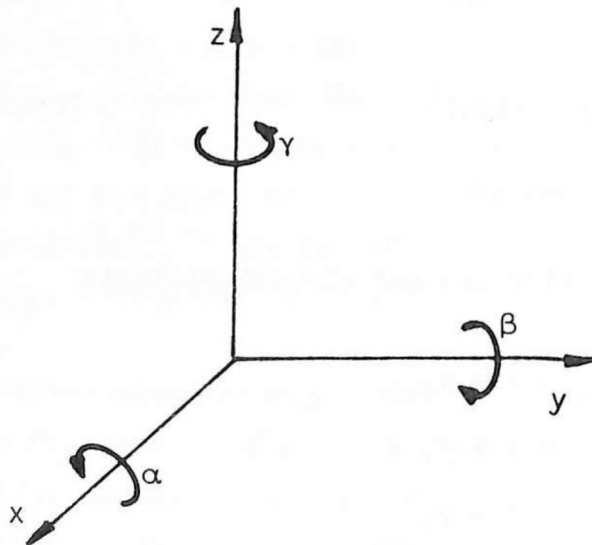
$$y_{VS} = y_S + y_V$$

$$z_{VS} = z_S + z_V$$

x_V, y_V, z_V : Verschiebungen eines Körperspunktes

3. Schritt: Rotation des Körpers

Der Körper wird zunächst um die x-Achse (Winkel α), dann um die y-Achse (Winkel β) und zuletzt um die z-Achse (Winkel γ) gedreht. Beachten Sie bitte die mathematisch positiven Drehrichtungen:



Drehwinkel in mathematisch positiver Richtung erhalten ein positives Vorzeichen, andere ein negatives Vorzeichen.

Do not sale !

Die Gleichungen für die Rotation sollen hier nicht hergeleitet werden. Sie sind, wie einige andere Gedanken zur 3D-Grafik auch, dem Buch von ENCARNACAO entnommen:

$$\begin{aligned}x_{rvs} &= x_{vs} * \cos \beta * \cos \gamma + \\ & y_{vs} * (-\cos \alpha * \sin \gamma + \sin \alpha * \sin \beta * \cos \gamma) + \\ & z_{vs} * (\sin \alpha * \sin \gamma + \cos \alpha * \sin \beta * \cos \gamma)\end{aligned}$$

$$\begin{aligned}y_{rvs} &= x_{vs} * \cos \beta * \sin \gamma + \\ & y_{vs} * (\cos \alpha * \cos \gamma + \sin \alpha * \sin \beta * \sin \gamma) + \\ & z_{vs} * (-\sin \alpha * \cos \gamma + \cos \alpha * \sin \beta * \sin \gamma)\end{aligned}$$

$$\begin{aligned}z_{rvs} &= x_{vs} * (-\sin \beta) + \\ & y_{vs} * \sin \alpha * \cos \beta + \\ & z_{vs} * \cos \alpha * \cos \beta\end{aligned}$$

4. Schritt: Projektion auf die x,y-Ebene

Wenn man die Parallelprojektion gewählt hat, dann kann man die z-Koordinate einfach vergessen:

$$\begin{aligned}x_{proj} &= x_{rvs} \\ y_{proj} &= y_{rvs}\end{aligned}$$

Bei der Zentralprojektion gehen die Koordinaten des Blickpunktes (x_z, y_z, z_z) mit in die Berechnungen ein. Dazu wird eine Hilfsgröße w definiert zu

$$w = 1 - \frac{z_{rvs}}{z_z}$$

Dann ergeben sich die zentralprojizierten Koordinaten als:

Do not sale !

$$x_{\text{proj}} = \frac{x_{\text{rvs}} - x_z}{w}$$

$$y_{\text{proj}} = \frac{y_{\text{rvs}} - y_z}{w}$$

Damit ist die 3D-Transformation im Prinzip beendet. Es kann aber sinnvoll sein, die Bildkoordinaten einer weiteren Manipulation zu unterziehen, um das projizierte Bild auf der Zeichenebene u, v darstellen zu können:

5. Schritt: Skalierung des Bildes

$$u_s = x_{\text{proj}} * u_f$$

$$v_s = y_{\text{proj}} * v_f$$

Damit wird das Bild auf die gewünschte Größe gebracht.

u_f, v_f : Skalierungs- (Vergrößerungs-) Faktoren des Bildes

6. Schritt: Verschiebung des Bildes

$$u = u_s + u_v$$

$$v = v_s + v_v$$

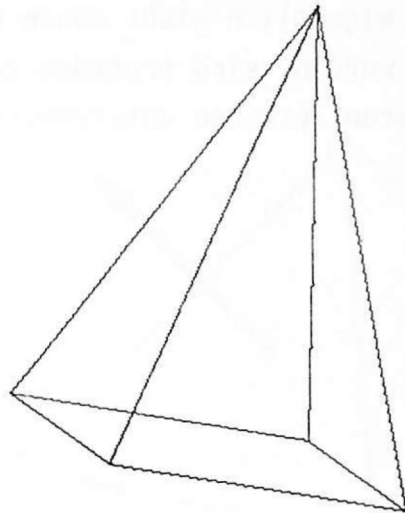
Damit wird das skalierte Bild auf der Bildebene verschoben.

u_v, v_v : Verschiebungen eines Bildpunktes

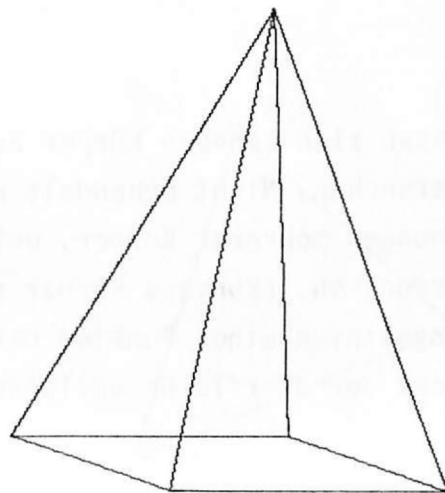
Mit diesem letzten Schritt ist die 3D-Transformation abgeschlossen.

Ein Programm, mit dem auf diese Art und Weise Körper 3D-transformiert und gezeichnet werden können, wird im Abschnitt 9.3 noch vorgestellt werden. Damit lassen sich Bilder wie folgende zeichnen:

Do not sale !



FAKTOR	:	1	1	1
VERSCHIEBUNG	:	0	0	0
WINKEL	:	-80	20	-10
BLICKPUNKT	:	0	0	0 P
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	70	-300	



FAKTOR	:	1	1	1
VERSCHIEBUNG	:	0	0	0
WINKEL	:	-80	30	0
BLICKPUNKT	:	0	0	0 P
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	110	-350	

Do not sale !

Bei diesen Bildern ist der dreidimensionale Eindruck noch nicht vollkommen: Körperkanten, die man eigentlich nicht sehen dürfte, da sie vom Körpervolumen verdeckt werden, sind trotzdem zu sehen. Wie man diese Kanten findet und deren Zeichnen unterdrückt, verrät der nächste Abschnitt.

9.2 DAS HIDDEN-LINE-PROBLEM

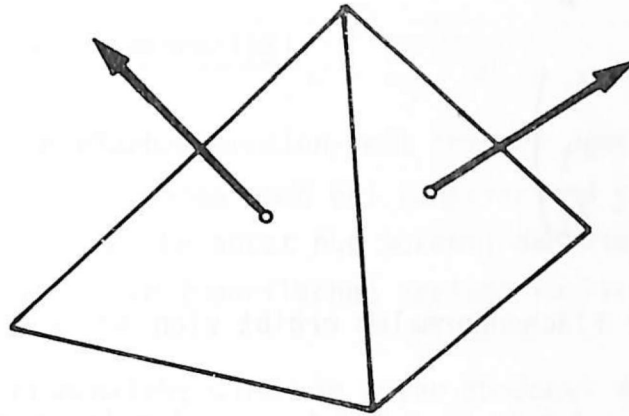
Ein ganz besonderes Problem der 3D-Grafik ist das Auffinden von unsichtbaren Körperkanten. Gesucht sind Algorithmen, die zuverlässig alle verdeckten Kanten finden, ohne daß die Rechenzeit zu groß wird. Beide Forderungen lassen sich nicht gemeinsam verwirklichen. Deswegen gibt es verschiedene Hidden-Line-Algorithmen, die sich nur bei ganz bestimmten Körpern anwenden lassen. Zwei dieser Algorithmen sollen hier vorgestellt werden.

9.2.1 Der Flächennormalen-Test

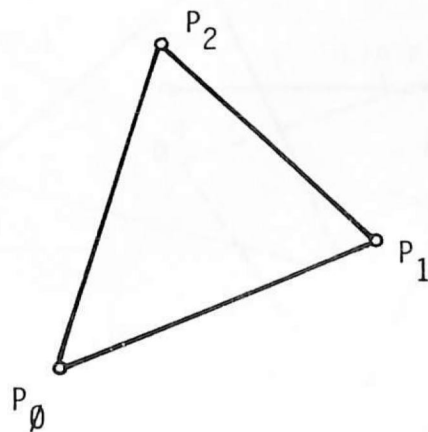
Mit dem Flächennormalen-Test lassen sich konvexe Körper auf Sichtbarkeit der Kanten hin untersuchen. Nicht behandelt werden können konkave Körper und Anordnungen mehrerer Körper, bei denen sich Körper gegenseitig verdecken. (Konvexe Körper sind Körper, bei denen jede Verbindungslinie eines Punktes der Oberfläche mit einem anderen Punkt der Oberfläche vollständig im Innern des Körpers liegt.)

Wir betrachten unser Pyramidenbeispiel. Das Bild auf der nächsten Seite zeigt, daß bei der gewählten Projektion drei Körperflächen unsichtbar und zwei sichtbar sind. Sichtbar sind die Oberflächen, deren Flächennormale (Senkrechte auf die Fläche) eine positive z-Komponente haben. D.h., diese Flächennormalen weisen (zumindest etwas) in Richtung der positiven z-Achse.

Do not sale !



Die Flächennormale läßt sich mit dem sogenannten "Vektorprodukt" berechnen. Gegeben sei eine Dreiecksfläche



Die Dreieckseiten lassen sich als Vektoren schreiben:

Do not sale !

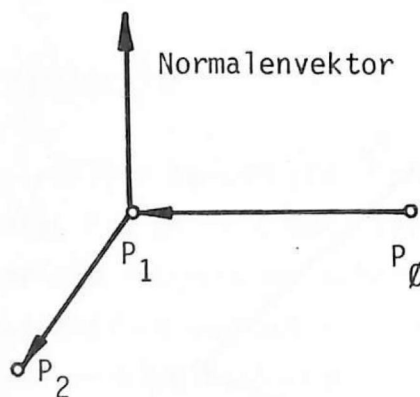
$$\vec{P_0 P_1} = \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \\ z_1 - z_0 \end{pmatrix}$$

$$\vec{P_1 P_2} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}$$

Die z-Komponente der Flächennormalen ergibt sich zu

$$z_{no} = (x_1 - x_0) * (y_2 - y_1) - (y_1 - y_0) * (x_2 - x_1)$$

Die sich mit dieser Gleichung ergebende Flächennormale steht immer so auf der Fläche, daß sie mit $\vec{P_0 P_1}$ und $\vec{P_1 P_2}$ ein Rechtssystem bildet:



Beim obigen Beispiel steht also die Normale so auf der Dreiecksfläche, daß sie zum Leser hin zeigt. Wenn man die Reihenfolge der Vektoren $\vec{P_0 P_1}$ und $\vec{P_1 P_2}$ vertauscht, dann zeigt die Normale vom Leser weg in das Buch hinein.

Beim Normalentest berechnet man also zu jeder Körperteiloberfläche die Flächennormale und kontrolliert, ob deren z-Komponente positiv

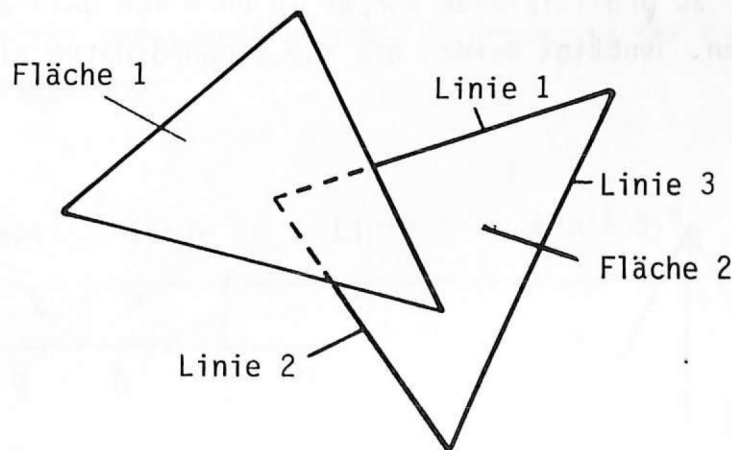
Do not sale !

oder negativ ist. Ist sie positiv, so ist die Fläche sichtbar. Wenn sie negativ ist, dann ist die Fläche unsichtbar.

9.2.2 Der Linien-Flächen-Test

Im Gegensatz zum Flächennormalen-Test ist der Linien-Flächen-Test universell anwendbar, also auch bei konkaven und sich gegenseitig verdeckenden Körpern. Er setzt nur voraus, daß die ganze Körperoberfläche in ebene Teiloberflächen zerlegt vorliegt.

Beim Linien-Flächen-Test wird wie zuvor zunächst der Körper in die Zeichenebene projiziert. Danach wird jede Körperkante mit jeder Körperteiloberfläche verglichen. Es wird kontrolliert, ob sich die Kante und die Oberfläche (auf der Bildebene) schneiden. Wenn das der Fall ist, dann wird der Schnittpunkt berechnet. Je nachdem, ob die Linie im Raum vor oder hinter der Fläche liegt, wird ein Teil der Linie verdeckt oder nicht.



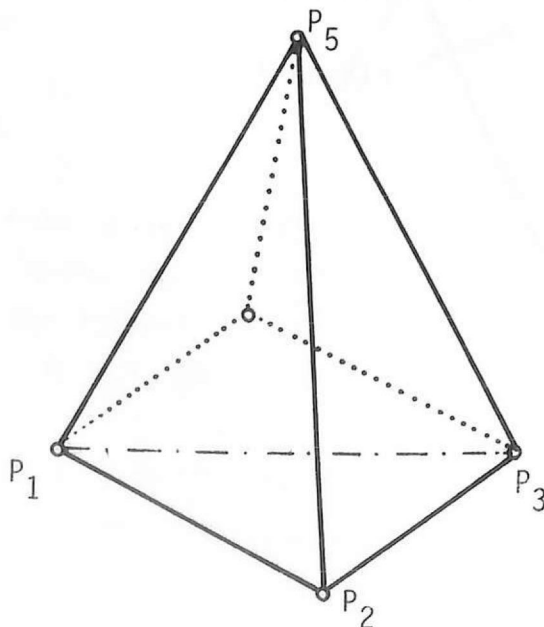
Jede Linie der Fläche 2 wird mit der Fläche 1 verglichen. Die Linien 1 und 2 werden geschnitten, die Linie 3 nicht. Da im Beispiel die Fläche 2 hinter der Fläche 1 liegen soll, werden die Linien 1 und 2 abgeschnitten.

Auf diese Weise wird in einem sehr zeitaufwendigen Verfahren jede Linie mit jeder Fläche verglichen.

9.3 EIN HIDDEN-LINE-PROGRAMM

Das nun folgende BASIC-Programm ermöglicht die 3D-Transformation von Körpern und die Suche der versteckten Linien nach dem Flächennormalen-Test. Es würde den Rahmen dieses Buches sprengen, wenn auch noch der Linien-Flächen-Test programmiert werden sollte. Es gibt aber ein fertiges PC-1500-Programm für diese Aufgabe (vergl. Literaturverzeichnis).

Zunächst muß der zu projizierende Körper in Form von DATA-Zeilen vorgegeben werden. Benötigt werden die x,y,z-Koordinaten aller Eckpunkte.



	x	y	z
P ₁	0	0	0
P ₂	100	0	0
P ₃	100	100	0
P ₄	0	100	0
P ₅	50	50	150

Do not sale !

Die Körperkanten werden durch Angabe von Anfangs- und Endpunkt bestimmt:

Kante	Anfangspunkt	Endpunkt
1	1	2
2	2	3
3	3	4
4	4	1
5	1	5
6	2	5
7	3	5
8	4	5
9	1	-3

Die Körperteiloberflächen werden durch die sie begrenzenden Kanten definiert. Dabei muß aus programmtechnischen Gründen die ganze Oberfläche in Dreiecke zerlegt sein. Bei der Pyramide ist aber die Grundfläche ein Quadrat. Dieses zerlegen wir in zwei Dreiecke mit der Zusatzlinie 9. Diese Hilfslinie soll natürlich nicht gezeichnet werden. Deswegen ist der Endpunkt negativ vorgegeben.

Dreieck	Linie 1	Linie 2	Linie 3
1	2	1	9
2	3	9	4
3	1	6	5
4	2	7	6
5	3	8	7
6	4	5	8

Bitte beachten Sie: Die Flächennormale muß in die Richtung zeigen,

Do not sale !

aus der die jeweilige Fläche sichtbar ist. Sie muß also von der Körperoberfläche wegzeigen. Die Dreieckslinien müssen deswegen entsprechend angeordnet werden (Rechtssystem!).

Das Programm liest zu Beginn diese ganzen Daten aus DATA-Zeilen. Es speichert die Punktkoordinaten in den X-, Y- und Z-Feldern, die Linienanfangs- und -endpunkte in den LA- und LE-Feldern und die Dreiecksseiten in den D1-, D2- und D3-Feldern. Nach der kompletten 3D-Transformation (wie im Abschnitt 9.1 beschrieben) wird gefragt, ob ein Normalentest durchgeführt werden soll. Wenn das der Fall ist, dann müssen irgendwie die unsichtbaren Linien gekennzeichnet werden. Dies geschieht so: Zu Beginn des Normalentests werden alle Linienanfangspunkte mit einem negativen Vorzeichen versehen (Zeile 8575). Wenn nun der Normalentest feststellt, daß eine Dreieckfläche sichtbar ist, dann werden die drei Dreiecksseiten von ihrem negativen Vorzeichen befreit (Zeile 8170 ... 8180). Später beim Zeichnen werden nur solche Linien gezeichnet, bei denen sowohl der Anfangs- als auch der Endpunkt positiv ist (Zeile 8780, 8785). Vor dem Zeichnen werden die minimalen und maximalen Bildkoordinaten U_{klein} , $U_{\text{groß}}$, V_{klein} , $V_{\text{groß}}$ bestimmt (Zeile 8595 ... 8645). Der Anwender erhält die Möglichkeit, über eine Bildskalierung und -verschiebung das projizierte Bild genau auf der Zeichenebene zu plazieren. Das Programm endet mit dem Ausdruck der eingegebenen Parameter.

Hier nun das Programmlisting und einige Bilder:

```
8000 "3D-TRANSFORMATION"  
8005 REM -----  
8010 REM  SKALIERUNG  
8015 REM -----  
8020 X=X*XF  
8025 Y=Y*YF  
8030 Z=Z*ZF  
8035 REM -----  
8040 REM  UERSCHIEBUNG  
8045 REM -----
```

Do not sale !

```
8050 X=X+XU
8055 Y=Y+YU
8060 Z=Z+ZU
8065 REM -----
8070 "ROTATION"
8075 REM -----
8080 XR=X*CB*CG+Y*(-CA*SG+SA*SB*CG)+Z*(SA*SG+CA*SB*CG)
8085 YR=X*CB*SG+Y*(CA*CG+SA*SB*SG)+Z*(-SA*CG+CA*SB*SG)
8090 ZR=X*(-SB)+Y*(SA*CB)+Z*(CA*CB)
8095 REM -----
8100 REM  PROJEKTION
8105 REM -----
8110 W=1:IF P$="Z"LET W=1-ZR/ZZ
8115 U=(XR-XZ)/W
8120 V=(YR-YZ)/W
8125 RETURN
8130 REM -----
8135 "NORMALENTTEST"
8140 REM -----
8145 U0=U(ABS LE(D1(I))) - U(ABS LA(D1(I)))
8150 V0=V(ABS LE(D1(I))) - V(ABS LA(D1(I)))
8155 U1=U(ABS LE(D2(I))) - U(ABS LA(D2(I)))
8160 V1=V(ABS LE(D2(I))) - V(ABS LA(D2(I)))
8165 NO=U0*U1-U1*U0
8170 IF NO>0LET LA(D1(I))=ABS (LA(D1(I)))
8175 IF NO>0LET LA(D2(I))=ABS (LA(D2(I)))
8180 IF NO>0LET LA(D3(I))=ABS (LA(D3(I)))
8185 RETURN
8190 REM -----
8195 "P"
8200 REM -----
8205 REM  3D-PROJEKTION
8210 REM -----
8215 WAIT 0
8220 INPUT "Externer Drucker ? (J/N) ";D$
8225 IF D$<>"J"LET D$=""
8230 IF D$<>" "SETCOM 1200,8,N,1:OUTSTAT 0:SETDEV PO
8235 IF D$<>" "CONSOLE 0,0
8240 IF D$=""GRAPH
8245 IF D$<>" "LPRINT CHR$(27)+"b"
8250 S=1:INPUT "CSIZE ? ";S
8255 REM -----
8260 REM  EINGABE DER FIGUR
8265 REM -----
8270 INPUT "Name der Figur ? ";N$
8275 RESTORE N$
```

Do not sale !

```
8280 READ P,L,D
8285 DIM X(P),Y(P),Z(P),U(P),V(P),LA(L),LE(L),D1(D),D2(D),D
3(D)
8290 FOR I=1TO P
8295 READ X(I),Y(I),Z(I)
8300 NEXT I
8305 FOR I=1TO L
8310 READ LA(I),LE(I)
8315 NEXT I
8320 FOR I=1TO D
8325 READ D1(I),D2(I),D3(I)
8330 NEXT I
8335 REM -----
8340 REM VOREINSTELLUNGEN
8345 REM -----
8350 XF=1:YF=1:ZF=1:XV=0:YV=0:ZV=0
8355 REM -----
8360 REM EINGABE : SKALIERUNG
8365 REM -----
8370 S$="":INPUT "Skalierung ? (J/N) ";S$
8375 IF S$="J"INPUT "XF = ? ";XF
8380 IF S$="J"INPUT "YF = ? ";YF
8385 IF S$="J"INPUT "ZF = ? ";ZF
8390 REM -----
8395 REM EINGABE : VERSCHIEBUNG
8400 REM -----
8405 U$="":INPUT "Verschiebung ? (J/N) ";U$
8410 IF U$="J"INPUT "XU = ? ";XU
8415 IF U$="J"INPUT "YU = ? ";YU
8420 IF U$="J"INPUT "ZU = ? ";ZU
8425 REM -----
8430 REM EINGABE : WINKEL
8435 REM -----
8440 A=0:INPUT "Winkel um x-Achse ? ";A
8445 B=0:INPUT "Winkel um y-Achse ? ";B
8450 G=0:INPUT "Winkel um z-Achse ? ";G
8455 SA=SIN A:SB=SIN B:SG=SIN G
8460 CA=COS A:CB=COS B:CG=COS G
8465 REM -----
8470 REM EINGABE : PROJEKTIONSART
8475 REM -----
8480 P$="P":INPUT "Projektionsart ? (P/Z) ";P$
8485 XZ=0:YZ=0:ZZ=0
8490 IF P$="Z"INPUT "Blickpunkt X-Wert ? ";XZ
8495 IF P$="Z"INPUT "Blickpunkt Y-Wert ? ";YZ
8500 IF P$="Z"INPUT "Blickpunkt Z-Wert ? ";ZZ:IF ZZ<=0GOTO
8500
```

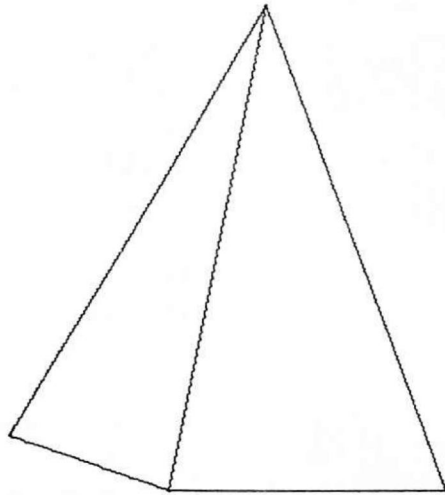
Do not sale !

```
8505 REM -----
8510 REM 3D-TRANSFORMATION ALLER PUNKTE
8515 REM -----
8520 FOR I=1 TO P
8525 PRINT "3-D"; I
8530 X=X(I):Y=Y(I):Z=Z(I)
8535 GOSUB "3D-TRANSFORMATION"
8540 U(I)=U:V(I)=V
8545 NEXT I
8550 REM -----
8555 REM NORMALENTTEST
8560 REM -----
8565 NO$="": INPUT "Normalentest ? (J/N) "; NO$
8570 IF NO$ <> "J" GOTO "P-NO"
8575 FOR I=0 TO L: LA(I)=-LA(I): NEXT I
8580 FOR I=0 TO D: PRINT "N-TEST"; I: GOSUB "NORMALENTTEST": NEXT
I
8585 REM -----
8590 "P-NO"
8595 REM -----
8600 REM BESTIMMUNG DER EXTREMA
8605 REM -----
8610 UK=U(1):UG=U(1)
8615 VK=V(1):VG=V(1)
8620 FOR I=2 TO P
8625 IF U(I)<UK LET UK=U(I)
8630 IF U(I)>UG LET UG=U(I)
8635 IF V(I)<VK LET VK=V(I)
8640 IF V(I)>VG LET VG=V(I)
8645 NEXT I
8650 REM -----
8655 REM EINGABE : BILDMANIPULATION
8660 REM -----
8665 USING "####"
8670 PRINT "U: "; UK; UG;
8675 UF=1: INPUT " U-Faktor ? "; UF
8680 CLS
8685 PRINT "U: "; UK*UF; UG*UF;
8690 UV=0: INPUT " U-Versch. ? "; UV
8695 CLS
8700 PRINT "U: "; UK; UG;
8705 UF=1: INPUT " U-Faktor ? "; UF
8710 CLS
8715 PRINT "U: "; UK*UF; UG*UF;
8720 UV=0: INPUT " U-Versch. ? "; UV
```

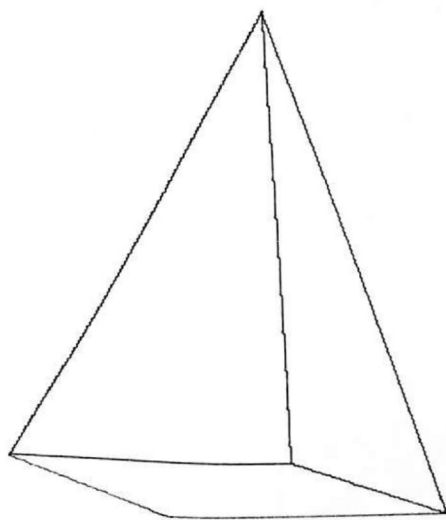
Do not sale !


```
8725 REM -----
8730 REM  BILDMANIPULATION
8735 REM -----
8740 FOR I=1TO P
8745 U(I)=U(I)*UF+UU
8750 U(I)=U(I)*UF+UU
8755 NEXT I
8760 REM -----
8765 REM  ZEICHNEN
8770 REM -----
8775 FOR I=1TO L
8780 IF LA(I)<0GOTO "Pnext"
8785 IF LE(I)<0GOTO "Pnext"
8790 IF D$=""GLCURSOR (U(LA(I)),U(LA(I)))
8795 IF D$<>" "LPRINT "M";U(LA(I));", ";U(LA(I))
8800 IF D$=""LINE -(U(LE(I)),U(LE(I))),0
8805 IF D$<>" "LPRINT "D";U(LE(I));", ";U(LE(I))
8810 "Pnext"NEXT I
8815 REM -----
8820 REM  TEXTAUSGABE
8825 REM -----
8830 UK=VK*UF+UU
8835 IF D$=""GLCURSOR (0,UK+20):TEXT :CSIZE 1:LF 5
8840 IF D$<>" "LPRINT "M0,";UK+20:LPRINT "A"
8845 IF D$<>" "LPRINT CHR$(27)+"?"+CHR$(860+S)
8850 USING "#####"
8855 LPRINT "FAKTOR           : ";XF;YF;ZF
8860 LPRINT "VERSCHIEBUNG      : ";XU;YU;ZU
8865 LPRINT "WINKEL              : ";A;B;G
8870 LPRINT "BLICKPUNKT           : ";XZ;YZ;ZZ;" " ;P$
8875 LPRINT "BILDFAKTOR            : ";UF;UF
8880 LPRINT "BILDVERSCHIEBUNG     : ";UU;UU
8885 END
```

Do not sale !



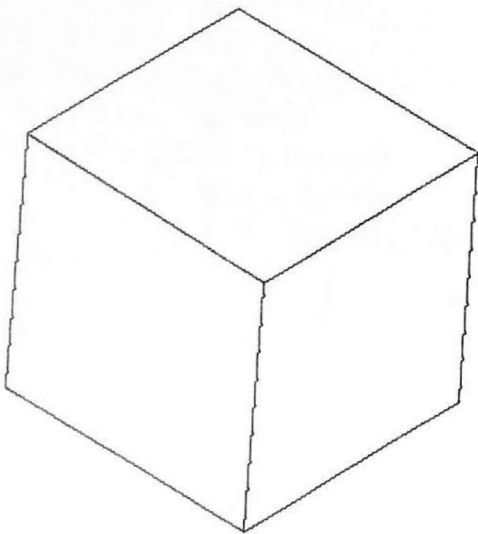
FAKTOR	:	1	1	1
VERSCHIEBUNG	:	0	0	0
WINKEL	:	-80	30	0
BLICKPUNKT	:	0	0	0 P
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	110	-350	



FAKTOR	:	1	1	1
VERSCHIEBUNG	:	0	0	0
WINKEL	:	-100	-30	0
BLICKPUNKT	:	0	0	0 P
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	10	-300	

Do not sale !

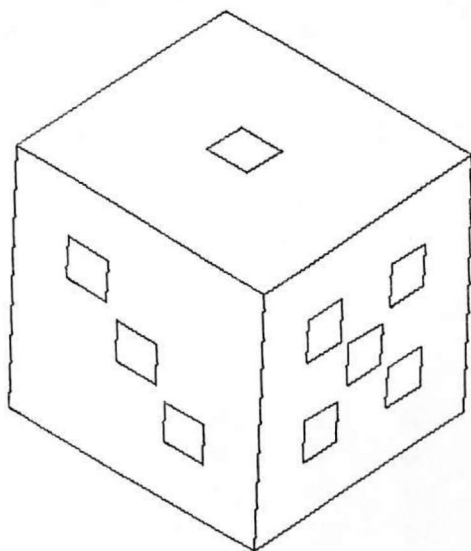
```
8890 "PYRAMIDE"  
8895 DATA 5,9,6  
8900 DATA 0,0,0  
8905 DATA 100,0,0  
8910 DATA 100,100,0  
8915 DATA 0,100,0  
8920 DATA 50,50,150  
8925 DATA 1,2  
8930 DATA 2,3  
8935 DATA 3,4  
8940 DATA 4,1  
8945 DATA 1,5  
8950 DATA 2,5  
8955 DATA 3,5  
8960 DATA 4,5  
8965 DATA 1,-3  
8970 DATA 2,1,9  
8975 DATA 3,9,4  
8980 DATA 1,6,5  
8985 DATA 2,7,6  
8990 DATA 3,8,7  
8995 DATA 4,5,8
```



```
FAKTOR          :    10    10    10  
VERSCHIEBUNG   :     0     0     0  
WINKEL         :     45    30   150  
BLICKPUNKT     :     0     0     0 P  
BILDFAKTOR     :     2     2  
BILDVERSCHIEBUNG :   300  -280
```

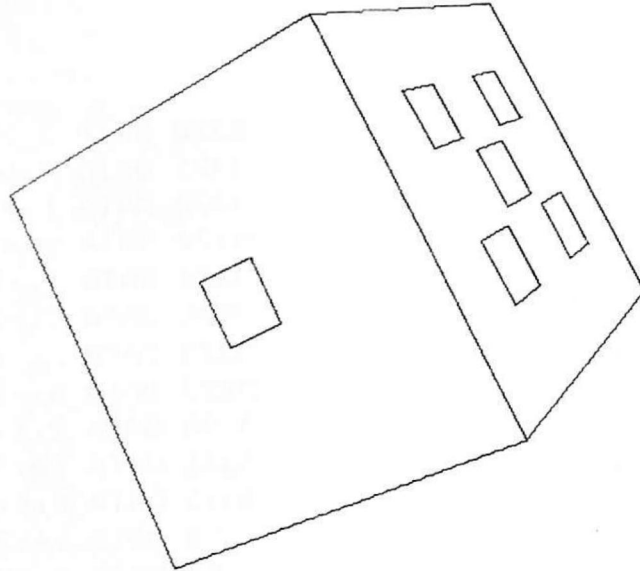
Do not sale !

9300	"WUERFEL"	9320	DATA 3,7
9301	DATA 8,18,12	9321	DATA 4,8
9302	DATA 0,0,0	9322	DATA 1,-3
9303	DATA 10,0,0	9323	DATA 5,-7
9304	DATA 10,10,0	9324	DATA 1,-6
9305	DATA 0,10,0	9325	DATA 2,-7
9306	DATA 0,0,10	9326	DATA 3,-8
9307	DATA 10,0,10	9327	DATA 4,-5
9308	DATA 10,10,10	9330	DATA 2,1,13
9309	DATA 0,10,10	9331	DATA 13,4,3
9310	DATA 1,2	9332	DATA 5,6,14
9311	DATA 2,3	9333	DATA 14,7,8
9312	DATA 3,4	9334	DATA 1,10,15
9313	DATA 4,1	9335	DATA 15,9,5
9314	DATA 5,6	9336	DATA 2,11,16
9315	DATA 6,7	9337	DATA 16,10,6
9316	DATA 7,8	9338	DATA 3,12,17
9317	DATA 8,5	9339	DATA 17,11,7
9318	DATA 1,5	9340	DATA 4,9,18
9319	DATA 2,6	9341	DATA 18,12,8

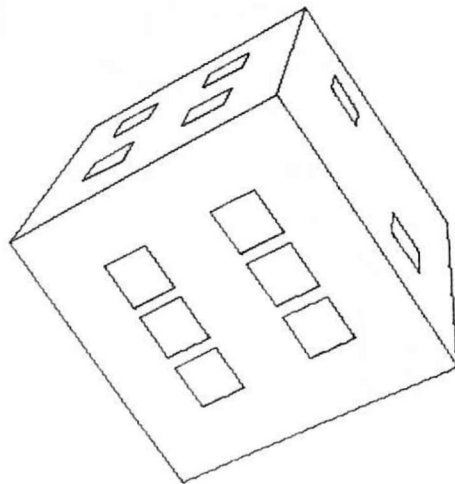


FAKTOR	:	10	10	10
VERSCHIEBUNG	:	0	0	0
WINKEL	:	45	30	150
BLICKPUNKT	:	0	0	0 P
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	300	-280	

Do not sale !

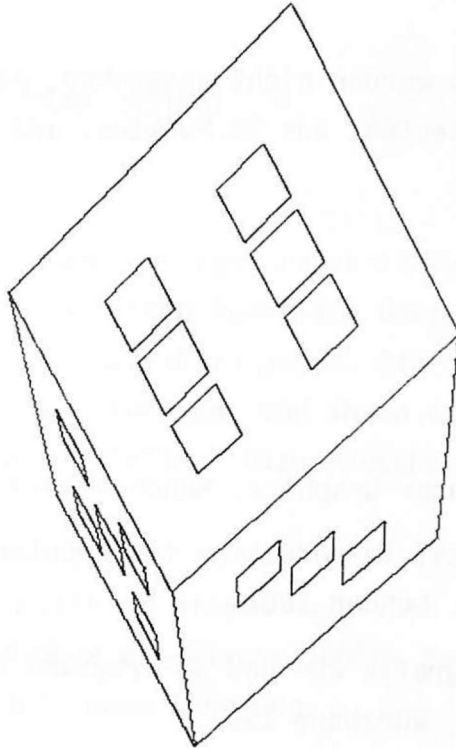


FAKTOR	:	10	10	10
VERSCHIEBUNG	:	0	0	0
WINKEL	:	10	20	210
BLICKPUNKT	:	0	0	300 Z
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	350	0	

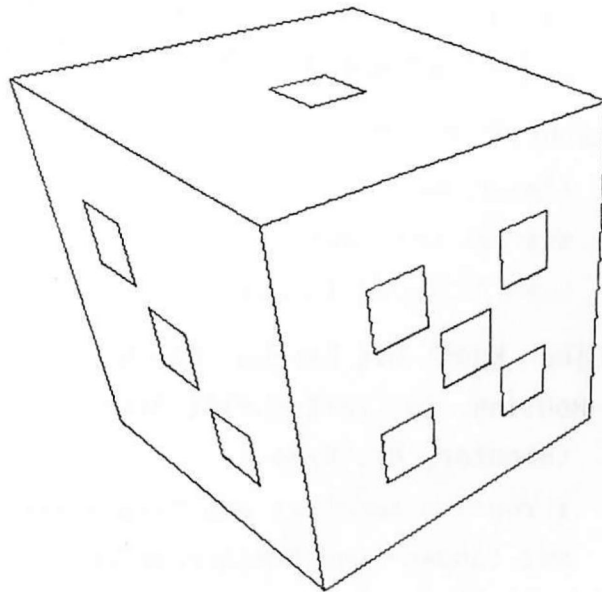


FAKTOR	:	10	10	10
VERSCHIEBUNG	:	0	0	0
WINKEL	:	10	200	210
BLICKPUNKT	:	50	50	350 Z
BILDFAKTOR	:	2	2	
BILDVERSCHIEBUNG	:	120	-50	

Do not sale !



FAKTOR : 10 10 10
VERSCHIEBUNG : 0 0 0
WINKEL : 10 200 210
BLICKPUNKT : -10 -600 500 Z
BILDFAKTOR : 2 2
BILDVERSCHIEBUNG : 0-1400



FAKTOR : 10 10 10
VERSCHIEBUNG : 0 0 0
WINKEL : 45 30 150
BLICKPUNKT : 0 0 300 Z
BILDFAKTOR : 2 2
BILDVERSCHIEBUNG : 400 -300

Do not sale !

Die DATA-Zeilen des Spielwürfels werden nicht angegeben, da es zu viele sind: Der Spielwürfel besteht aus 92 Punkten, 123 Linien und 54 Dreiecken.

LITERATUR:

- : ENCARNACAO, Jose L.: Computer Graphics. München 1975
- : ANGELL, Ian O.: A Practical Introduction to Computer Graphics. London 1981
- : POMASKA, Günter: Computergrafik 2D- und 3D-Programmierung 1. Aufl.; Würzburg 1984

PROGRAMM-LISTINGS FÜR PC-1500 (A):

- : 3D-PLOTTER. In: PC-1500 Programmier- und Programmhandbuch. Fischel GmbH, Berlin 1984
(Programm erlaubt die grafische Darstellung einer Funktion von zwei Variablen)
- : GLOBUS. In: Zeitschrift HC, Nr. 9/84
(Programm zeichnet ein Bild der Erdkugel mit den Umrissen der Kontinente. Die Erde kann beliebig im Raum gedreht werden.)
- : 3D-Darstellung einer Kugel mit Längen- und Breitenkreisen mit Hidden-Line-Routine. In: Zeitschrift Mikro + Klein - Computer, Nr. 6/84
(Programm zeichnet ein Bild einer Kugel mit Längen- und Breitenkreisen. Im Aufsatz wird interessante Information zum Auffinden der verdeckten Linien gegeben.)

Do not sale !

10 ZUSAMMENFASSUNG

Wir sind am Ende unseres Grafik-Lehrbuches angelangt. Ich habe versucht, Ihnen einen Überblick über die wichtigsten Verfahren der 2D- und 3D-Grafik zu geben. Die hier vorgestellten Programme sollen Sie dazu anregen, von ihnen ausgehend neue und bessere Programme zu erstellen. Dazu wünsche ich Ihnen viel Freude und Erfolg.

Falls Sie Kritik an diesem Buch üben wollen, dann lassen Sie mir Ihren Brief über die Fischel GmbH, Berlin zugehen. Ich freue mich über jede neue Anregung.

Kaiserslautern, im April 1985

Klaus Schreiner

Do not sale !

MEIN ANGEBOT AN PROGRAMMEN FÜR DEN

PROGRAMM-KASSETTEN FÜR PC-1500 (A):

(Erhältlich bei der Fischel GmbH, Berlin)

- : 3D-HIDDEN LINE *(Dies ist mein Programm zur 3D-Transformation und zur Berechnung der verdeckten Linien von Körpern. Als Hidden-Line-Algorithmus wird der Linien-Flächen-Test verwendet. Das Programm ist sehr komfortabel und erlaubt auch die Verwendung des CE-515P/516P!)*
- : WELTKUGEL *(Das Programm erlaubt die Darstellung der Erdkugel. Die Erde kann beliebig im Raum gedreht werden.)*

V E K M A T

VEKMAT ist ein Maschinenspracheprogramm zur Verarbeitung von eindimensionalen Feldern (Vektoren) und von zweidimensionalen Feldern (Matrizen). BASIC ist zwar in der Lage, mathematische Operationen mit Vektoren und Matrizen durchzuführen; es benötigt dazu aber wesentlich mehr Rechenzeit als VEKMAT. Die Möglichkeiten von VEKMAT sind folgende:

- : Addition, Subtraktion, Multiplikation und Division von Vektoren und Matrizen mit Zahlen
- : Addition und Subtraktion zweier Vektoren oder Matrizen
- : Multiplikation zweier Vektoren zum Skalarprodukt
- : Multiplikation zweier Matrizen

Das Besondere an VEKMAT ist neben der schnellen Rechengeschwindigkeit die Syntax der VEKMAT - Befehle. Die mathematische Aufgabe wird BASIC-ähnlich in Textvariablen gespeichert. Beim Aufruf von VEKMAT wird einfach angegeben, in welcher Textvariablen die Aufgabe steht.

Lieferumfang:

- : Kassette mit VEKMAT
- : Ausführliche Bedienungsanleitung
- : Hinweise für Maschinensprache-Neulinge

Größe: 650 Bytes

Preis: DM 25.-

Do not sale !

DATAGRAM

DATAGRAM ist ein vollständig in der Maschinensprache des PC-1500 (A) geschriebenes Datenbankprogramm. Folgende Möglichkeiten bietet es:

- : Erstellen und Verwalten einer Datenbank
- : Ändern von einzelnen Eingaben dieser Datenbank
- : Löschen von Daten
- : Suchen nach Begriffen
- : Ordnen der Datenbank nach bestimmten Kriterien
- : Abspeichern und Laden der Datenbank auf/von Kassette
- : Warm- und Kaltstart
- : Ausdruck der Datenbank oder von Teilen davon auf dem CE-150
- : Verwalten einer zweiten Datenbank im 2. Speicherblock des PC-1500 (A), falls dieser Speicherbereich ausgebaut ist

Besonderheiten:

- : DATAGRAM ist vollständig relokativ
- : DATAGRAM ist vollständig menügesteuert
- : Durch spezielle Kontrollunterprogramme wird der Rechnerabsturz verhindert (sehr wichtig bei Maschinenspracheprogrammen)
- : Falls der Rechner wegen erschöpfter Batterien des CE-150 ohne Vorwarnung mit ERROR 78/80 ansteigt, bleibt die Datenbank erhalten
- : Benutzern des TRAMSOF T00L 2 (FSAVE/FLOAD) wird die Möglichkeit zur schnellen Datensicherung geboten
- : Dito für die Fast-Programme von J. Jürgens und A. Pfründer (S-Tape)

Lieferumfang:

- : Kassette mit DATAGRAM
- : 16-seitige Bedienungsanleitung in bekannter Qualität
- : Spezielle Hinweise für Anwender, die sich mit Maschinensprache nicht auskennen

Größe: 5.5 kByte

Preis: DM 98.-

3D - HIDDEN LINE

Es gibt viele PC-1500 (A) - Programme zur Darstellung von dreidimensionalen Körpern mit dem CE-150, aber nur ein 3D-HIDDEN LINE. Mit 3D-HIDDEN LINE werden Körper nicht nur auf die Zeichenebene projiziert, sondern sie werden auch auf die Existenz von nicht sichtbaren Teilen (Hidden Lines) hin untersucht.

Im einzelnen kann das Programm folgendes:

- : 3D-Projektion eines zuvor eingegebenen Körpers
- : Manipulation der Körperdaten (Skalierung, Verschiebung)
- : Parallel- und Zentralprojektion
- : Manipulation des projizierten Bildes (Skalierung, Verschiebung)
- : Aufsuchen der versteckten Linien
- : Ausgabe des Bildes auf dem CE-150 und den Grafik-Plottern CE-515P und CE-516P
- : Das Programm ist vollständig menügesteuert

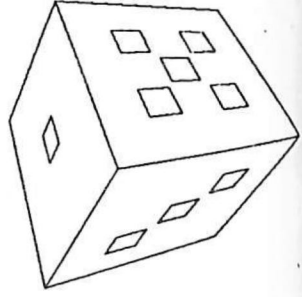
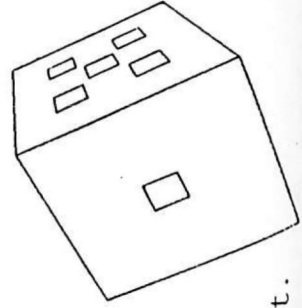
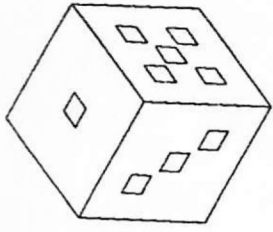
Informatiker sind immer wieder erstaut darüber, daß man mit dem "kleinen" PC-1500 (A) sogar Hidden Line-Probleme behandeln kann!

Lieferumfang:

- : Kassette mit 3D-HIDDEN LINE
- : 20-seitige Bedienungsanleitung

Größe: 10 kByte

Preis: DM 40.-



Do not sale !

Alle Preise incl. 14 Prozent Mwst.

Neue, cursorgesteuerte Menüwahl.

Nach Eingabe der Parameter wie Funktion, Wertebereich (mit Ausschnittvergrößerung), Betrachtungswinkel, Farbe und Anzahl der Gitterlinien (bis 80 Linien in x- und y- Ebene möglich), wird eine 3D- Graphik von hoher Auflösung und bisher nicht gefundener Exaktheit geplottet.

Als weitere Neuheit kann die sichtbare Unterseite der Graphik auch in einer anderen Farbe dargestellt werden.

Durch Fehlerbehandlungsroutinen, ein neuartiges Logik-Modul zur Move- und Plotabfrage, sorgfältiges Austesten und eine ausführliche Bedienungsanleitung ist dieses Programm für den MZ- 731 bisher konkurrenzlos und ein Muß für jeden 3D- Anwender.

Kassettenversion DM 50.- incl. 14% Mwst.

Sehr geehrte Damen und Herren,

Sie sind auf dem Gebiet der Computer-Software tätig und werden sicher oft mit Fragen des Software-Rechts konfrontiert. Bei der Lösung solcher Probleme ist unser Buch

SOFTWARE-RECHT

Die Bestimmungen des Urheber- und Wettbewerbsrechts für Computer-Programme (Dr. Roger Dorsch/Bernd Fischel, ISBN 3-924327-03-3, ca. 120 S. DIN A5, 29.00 DM incl. 7 % Mwst/Versand)

eine unschätzbare Hilfe. Es behandelt bzw. beantwort folgende Fragen:

1. Was bedeutet der Copyright-Vermerk?
2. Weshalb lassen sich manche Programme nicht kopieren?
3. Ist es erlaubt, ausländische Software ins Deutsche zu übersetzen?
4. Zu welchem Zweck darf man ein Programm kopieren?
5. Wer darf Software verkaufen?
6. Kann man sich ein Programm patentieren lassen?
7. Wer informiert über Gerichtsurteile und den Stand der Rechtsprechung?
8. Wie kann man sich vor Raubkopien schützen?
9. Worin besteht der Urheberrechtsschutz für Software?
10. Welche Behörden, Gerichte und Anwälte sind im Streitfall zuständig?
11. Welche Ansprüche lassen sich durchsetzen?
12. Welche Musterprozesse sind entschieden?
13. Welche Vereinbarungen soll ein Lizenzvertrag enthalten?
14. Welche Rechte und Pflichten ergeben sich für den Arbeitgeber und Arbeitnehmer?

Sichern Sie sich Ihr(e) Exemplar(e) von "Software-Recht" noch heute durch Einzahlung auf unser Konto (Bearbeitung nach Zahlungseingang), Einsendung eines Schecks oder Bestellung per Nachnahme. Mengenrabatte teilen wir Ihnen auf Anfrage mit.

Mit freundlichen Grüßen

gez. B. Fischel

PS: Wir führen auch Sharp-Microcomputer incl. Zubehör sowie den Brother Printer EP-44 (Schreibmaschine, Schönschriftdrucker und Terminal in einem Gerät) und den mm-Akustikkoppler AK-300 (FTZ-Nr., Batterie- und Netzbetrieb, 300 Baud, voll duplex, Originator- und Answermodus). Weitere Informationen auf Anfrage.

Do not sale!

A B O N N E M E N T

Wenn es Ihnen Spaß gemacht hat, diese Ausgabe von "Alles für Sharp Computer" zu lesen, und Sie sich auch in Zukunft durch unsere interessante Zeitschrift über alles Wissenswerte zum Thema Sharp Computer informieren wollen, dann sollten Sie nicht länger zögern, "Alles für Sharp Computer" jetzt im regelmäßigen Bezug per Post zu bestellen. Sichern Sie sich eine lückenlose Information und schicken Sie den Bestellabschnitt am besten noch heute ab. "Alles für Sharp Computer" kommt dann regelmäßig jeden Monat ins Haus, ohne daß Ihnen zusätzliche Kosten entstehen.

Alles für
SHARP
Computer

Bestellschein

Bitte vollständig und lesbar ausfüllen,
unterschreiben und einsenden an Fischel GmbH,
Kaiser-Friedrich-Str. 54a, D-1000 Berlin 12

- Ich abonniere die Zeitschrift "Alles für Sharp Computer" von der nächsten erreichbaren Ausgabe an (Preis pro Jahr 72 DM, Ausland 84 DM, Luftpostzuschlag 12 DM).
- Ich abonniere die Zeitschrift "Alles für Sharp Computer" von der Ausgabe ... (Monat) ... (Jahr) an (Preis pro Jahr 72 DM, Ausland 84 DM, Luftpostzuschlag 12 DM).

Das Abonnement verlängert sich um ein Jahr zu den dann jeweils gültigen Bedingungen, wenn es nicht 2 Monate vor Ablauf schriftlich gekündigt wird.

- Ich bestelle folgende schon erschienene Exemplare von "Alles für Sharp Computer" (Stückpreis 6 DM, Ausland 7 DM):
Heftnr.: ... , ... , ... , ... , ...

Alle Preise incl. 7 % Mwst.

Der Gesamtbetrag von DM

- liegt bar bei
- liegt als Verrechnungsscheck bei (schnellste Erledigung)
- wurde am auf das Postgirokonto der Fischel GmbH, Kontonr. 461533-103, BLZ 10010010, Postgiroamt Berlin überwiesen (Bearbeitung nach Zahlungseingang)
- liegt (nur bei kleineren Beträgen) in Briefmarken oder internationalen Antwortscheinen bei.

Name, Vorname

Straße

PLZ/Ort

Datum, Unterschrift

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung. Ich bestätige dies durch meine zweite Unterschrift.

Do not sale!

ISBN 3-924327-04-1

Do not sale !