

3. Stück

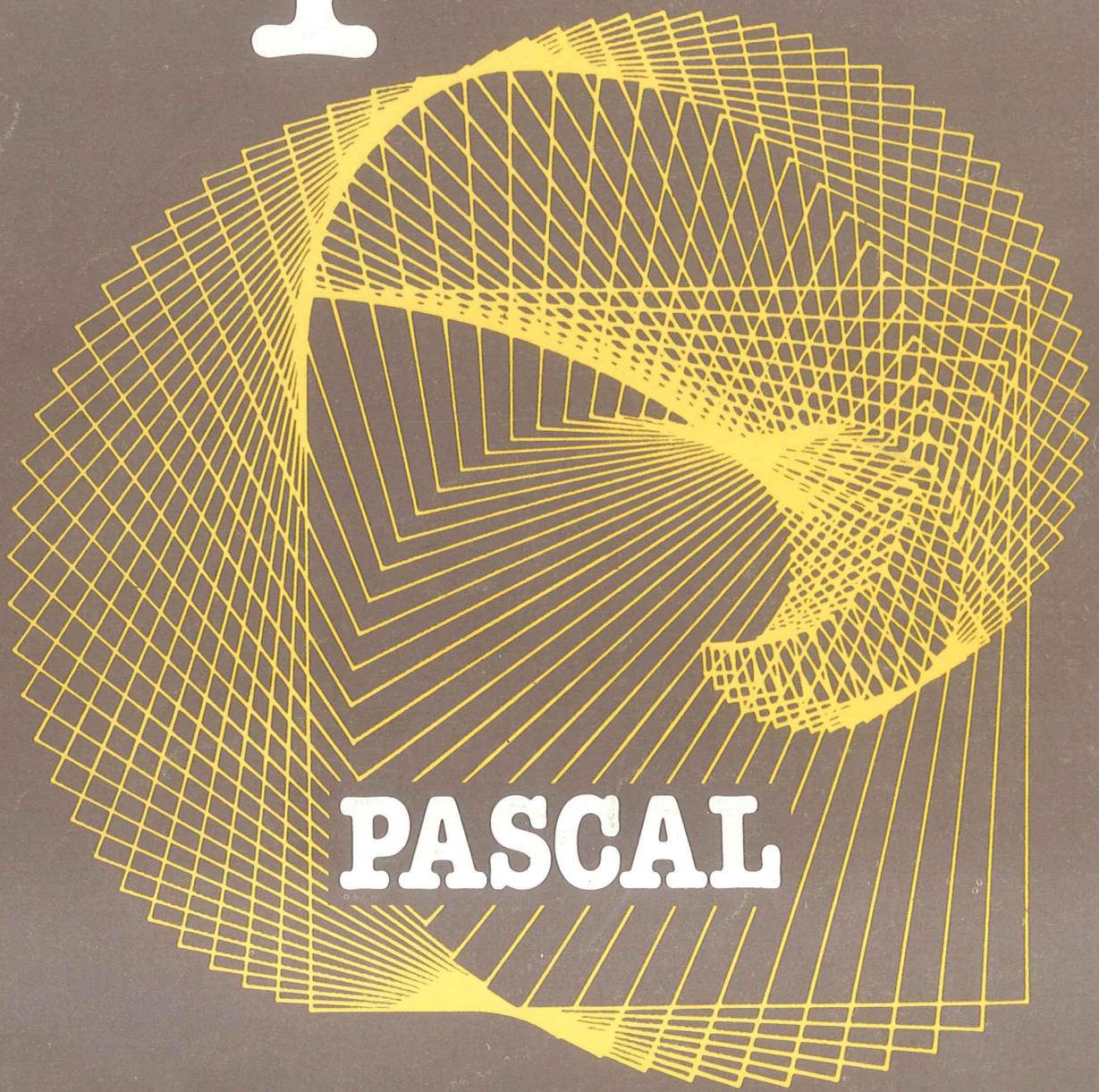
edv aspekte

4/85

Herausgegeben
von der Redaktion
rechentechnik
datenverarbeitung
DDR 5.00M

~~Hochschule für Verkehrswesen
„Friedrich List“
Sektion Mathematik, Rechentechnik
und Naturwissenschaften
WF Rechentechnik - Rechenzentrum~~

ingültig 19.7.93



PASCAL

Anwendungen für Wirtschaft · Wissenschaft · Technik

Inhalt

Erstes Programm (1)*	2	Werbung (46)	21	Postleitzahlen (77)	43
Druck einer Zeichenkette (2)	2	Darstellung einer Schwingung (42)	22	Anwendung der Strategie „LIFO“ (78)	43
Zeichenkettenkonstante (4)	2	Bildauswertung (41)	22	Dynamische Variable (79)	44
Grafik einer Schachfigur (3)	2	Alphabet (44)	23	Stücklistenzusammenstellung (80)	44
Lange Zeichenfolge (5)	3	Elementare Funktionen (45)	23	Häufigkeitsanalyse von Vornamen (81)	45
Beispielsprichwort (6)	3	Zweierpotenzen (48)	24	Protokoll einer Häufigkeitsanalyse (82)	46
Syntax eines Programms (7)	3	Sollwertabweichung (47)	24	Optische Täuschungen (83)	47
Konstante (8)	4	Werkzeugverschleißnorm (49)	25	Rechtecke im Raum (84)	50
Rechenoperationen mit Konstanten (9)	4	Grafik gedämpfter Schwingungen (50)	26	Erzeugung eines Fächers (85)	50
Kreis-Berechnungen (10)	5	Arbeitstage (51)	26	Sinusapproximation (86)	50
Getriebebezeichnungen (11)	5	Zwei Schaltungen (52)	27	Graphische Darstellung einer Sinusfunktion (87)	51
Inventur (12)	6	Operationen mit eindimensionalen Feldern (53)	28	Zeichnung einer Flagge (88)	51
Kugel (14)	6	Vierpolmatrizen (54)	28	Zeichne Quadrate (89)	52
Berechnungen mehrerer Kreise (13)	6	Inverse elementare Funktionen (57)	29	Penrosches Dreieck (90)	52
Plan-Abrechnung (16)	7	Darstellung einer Matrix (55)	30	Ein nicht möglicher Körper (91)	53
Buchse (15)	8	Tabellierung von Größen des Kreises (56)	30	Verschachtelte Ellipsen (92)	53
Demonstration ganzer Zahlen (17)	8	Haus (59)	31	Gedrehte Ellipsen (93)	54
Beliebige Kreise (18)	9	Anwendung einer Marke (60)	31	Artikelliste (94)	54
Operationen mit logischen Größen (19)	9	Rationalisierung der Programmierung (58)	32	Analyse von Pseudozufallszahlen (95)	55
Rechteck mit Strich (21)	10	Dreiecksflächen (61)	32	Demonstration eines Seiteneffektes (96)	56
Druck paralleler Strecken (20)	10	Vergleich zweier Algorithmen (62)	33	Seiteneffekt bei der Multiplikation (97)	56
Schnittgeschwindigkeiten (23)	11	Übersicht über Standard-Prozeduren/Funktionen (63)	33	Verzeichnis der Programme (98)	56
Abstand zweier Punkte (25)	11	Einführung der Vektorrechnung (65)	34	Verzeichnis der Programme und Aufgabenstellungen (99)	58
Viereck drucken (22)	12	Umkehrung einer Zeichenkette (66)	35	Sachwortverzeichnis zu PASCAL-Programmen (100)	60
Berechnung der Hypotenuse (24)	13	PASCAL-Standardbezeichner (64)	35	Test einer FOR-Anweisung (101)	64
Übertragung von Text (29)	13	Gäste am Tisch (67)	36		
Summe von Produkten (26)	13	Gegenseitige Rekursion (68)	37		
Netz eines Tetraeders (27)	14	Kopie des Inhaltes von Lochkarten (69)	37		
Einführung von Vergleichsoperationen (31)	14	Hierarchie der Datentypen (70)	37		
Skalares Produkt (28)	15	Komplexe Arithmetik (72)	38		
Lage eines Punktes (32)	15	Iterative Berechnung des arithmetischen Mittels (71)	39		
Manipulation mit Worten (30)	16	Teilnehmerliste (73)	39		
Signifikante Stellen (34)	16	Mitgliederverzeichnis (74)	40		
Qualitätskontrolle (33)	17	Künstlerdatei (76)	41		
Nullteiler (35)	17	Kurbelwellen (75)	42		
Masse-Berechnung eines Rohteils (36)	18				
Zeichentabelle (39)	18				
Druck reeller Größen (37)	19				
Schachbrett-Muster (40)	20				
Nachbildung einer Schaltung (38)	20				
Boolesche Größen (43)	21				

*Die Zahlen in Klammern beziehen sich auf die Nummer des Programms entsprechend den Listen der Aufgaben Inhalt und Register.

edv aspekte

4. Jahrgang 4/1985

Verlag Die Wirtschaft Berlin
1055 Berlin, Am Friedrichshain 22
Verlagsdirektor: Dieter Grüneberg

edv-aspekte
Zeitschrift für spezielle Themen der Informationsverarbeitung, herausgegeben von der Redaktion rechen-technik/datenverarbeitung, 1055 Berlin, Am Friedrichshain 22
Chefredakteur: Franz Loll 4 38 73 41
Redakteur: Hans Weiß 4 38 73 16
Sekretariat: 4 38 72 33
Fernschreiber: 114 566
Titelgestaltung: Marlies Hawemann

Redaktionsschluß: 28. August 1985

Lizenz des Presseamtes beim Vorsitzenden des Ministerrates der DDR Nr. 1529

edv-aspekte
Erscheinungsweise vierteljährlich zum Bezugspreis DDR 5,00 M je Heft
EDV-Artikel-Nr. 1331
Auslandspreise sind dem Zeitschriftenkatalog des Außenhandelsbetriebes Buchexport zu entnehmen.

Satz: Verlag Die Wirtschaft, Berlin
Druck: (140) „Neues Deutschland“, Berlin

Anzeigenverwaltung:
Berliner Verlag, 1056 Berlin,
Karl-Liebknecht-Str. 29, Telefon: 2 70 33 02

Anzeigenannahme:
Berliner Verlag und Annahmestellen in Berlin und in den Bezirken
Zur Zeit gültige Anzeigenpreisliste Nr. 12

Im Ausland:
INTERWERBUNG GmbH – Gesellschaft für Werbung und Auslandsmessen der DDR, 1157 Berlin, Hermann-Duncker-Str. 89

Bestellungen nehmen entgegen:
Für die DDR
Sämtliche Postämter, der örtliche Buchhandel und der Verlag Die Wirtschaft Berlin
Inkasso-Zeitraum: vierteljährlich

Im Ausland:
In den sozialistischen Ländern nur der zuständige Postzeitungsvertrieb. In allen anderen Staaten der örtliche Buch- und Zeitschriftenhandel. Bestellungen des Buch- und Zeitschriftenhandels sind zu richten an

BUCHEXPORT
Volkseigener Außenhandelsbetrieb der DDR, DDR – 7010 Leipzig, Leninstr. 16, Postfach 160 oder an Verlag Die Wirtschaft, Berlin
DDR – 1055 Berlin, Am Friedrichshain 22

Mitglieder des Redaktionsbeirates
Dr. Claus Goedecke · Dr. Rolf Gräßler
Prof. Dr. sc. Gerhard Keßler · Dr. Rolf Kilian
Hans Kunau-Walter Münch · Axel Rathsack
Prof. Dr. sc. Wolfgang Schoppa (Vorsitzender)
Dr. Werner Schulze · Horst Stoll
Prof. Dr. Franz Stuchlik · Dr. Dieter Urban

Eine Unterstützung für den PASCAL-Nutzer

Immer mehr – vor allem junge – Menschen haben das Bedürfnis, sich in die Anwendung der Informatik einzuarbeiten und in diesem Zusammenhang die Programmierung von Rechnern zu erlernen. Dieser Lernvorgang kann mit Programmiersprachen wesentlich unterstützt werden. Die Sprache PASCAL wurde – im Unterschied zu anderen Programmiersprachen – auch unter methodischen Gesichtspunkten geschaffen und bewährt sich außerordentlich gut in der Lehre. Auch aus diesem Grund sind für viele Typen von Mikrorechnern PASCAL-Compiler verfügbar, die den genannten Prozeß wirksam unterstützen und eine Rechnernutzung im Dialog gestatten.

Das Anliegen dieser Sammlung von PASCAL-Beispielen besteht darin, die Einarbeitung in die Sprache PASCAL und vor allem deren Anwendung zu unterstützen. Deshalb wurden nicht nur für alle Programme die zugehörigen Resultate angegeben, sondern die Quelltexte auch mit ausführlichen Kommentaren versehen. Zugleich werden jeweils einleitend die Sprachelemente aufgelistet, deren Nutzung im Mittelpunkt des Programms steht, um so auch dessen Auswahl zu unterstützen. Diesem Auswahlvorgang sowohl durch Lernende als auch Lehrende dienen ein Verzeichnis aller Programme mit Kurzcharakteristiken ihrer Inhalte, ein ausführliches Sachwortregister, welches vor allem auf Bezeichnungen der Sprachelemente basiert und eine Tabelle, die den Programmen die entsprechenden Aufgabenstellungen zuordnet. (Weitergehende Benutzungshinweise siehe III. Umschlagseite.)

Die Themen wurden so gewählt, daß ein möglichst breiter Kreis von Problemen nicht nur der Informatik und ihrer Anwendungen, sondern auch anderer Gebiete behandelt wird. So beschäftigen sich die Programme zum Beispiel mit graphischen Darstellungen, der Listentechnik, dem Sortieren u. a. m. Mit einer entsprechenden Gestaltung der Drucklisten wird das Durcharbeiten der Aufgaben und Programme unterstützt. Die meisten Aufgabenstellungen wurden aus zum Teil sehr umfangreichen Problemstellungen der Praxis, das heißt aus Wirtschaft, Wissenschaft und Technik (CAD/CAM), abgeleitet und so vereinfacht, daß sie im Rahmen einer Einführung genutzt werden können. Vorliegendes Material ist als Ergänzung zu Lehrbüchern über PASCAL konzipiert. Eine Aneignung dieser Sprache kann aber nur durch aktive Auseinandersetzung mit den Aufgaben und Programmen erreicht werden, das heißt, mittels Modifikation der gewählten Lösungen und einem Vergleich der damit erzielten Ergebnisse mit den vorliegenden. Gleichzeitig sollte die Möglichkeit genutzt werden, vor allem die einführenden Programme mit anderen Sprachelementen zu schreiben und sie sorgfältig zu testen. Bereits kleine Mikrorechner verfügen über die geräte- bzw. programmtechnischen Voraussetzungen, um grafische Darstellungen zu ermöglichen. Deshalb wurden bei der Auswahl auch solche Aufgaben berücksichtigt, die sich auf das genannte Zubehör stützen. Die Reihenfolge der Aufgaben berücksichtigt methodische Aspekte und geht davon aus, daß der Leser die Programme auf einem Rechner abarbeiten läßt. Abschließend möchte ich Herrn Dipl. Math. B. Goes für seine Unterstützung bei der Druckaufbereitung recht herzlich danken.

Prof. Dr. Franz Stuchlik

Inventur

IN EINEM LAGER MIT DER BEZEICHNUNG 'XYZ' IST EINE INVENTUR DURCHZUFUEHREN. ES IST EIN PROTOKOLL DIESER INVENTUR ZU DRUCKEN.

```
PROGRAM INVENTUR(OUTPUT);
(* ARBEIT MIT VARIABLEN VOM TYP INTEGER *)
(* VARIABLENDEKLARATION, AUSDRUCK, ANWEISUNG *)
CONST
ARTIKEL=' SENKKOPFSCHRAUBEN';
TYP=' VOM TYP ';
STRICH='-----';
DOSTRICH='=====';
KUSTRICH='-----'; M='MAGDEBURG, DEN';
VAR
ANZAHL, SUMME: INTEGER;
BEGIN
WRITELN('P R O T O K O L L D E R');
WRITELN('I N V E N T U R I M L A G E R X Y Z');
WRITELN(STRICH, STRICH, STRICH); WRITELN(' ');
SUMME:=0; (* ANFANGSWERTZUWEISUNG AN VARIABLE *)
ANZAHL:=170;
WRITELN(ANZAHL, ARTIKEL, TYP, '123');
SUMME:=SUMME + ANZAHL;
ANZAHL:=225;
WRITELN(ANZAHL, ARTIKEL, TYP, '125');
SUMME:=SUMME + ANZAHL;
ANZAHL:=73130;
WRITELN(ANZAHL, ARTIKEL, TYP, '135');
SUMME:=SUMME + ANZAHL;
ANZAHL:=33130;
WRITELN(ANZAHL, ARTIKEL, TYP, '150');
SUMME:=SUMME + ANZAHL;
WRITELN(KUSTRICH); WRITELN(SUMME, ARTIKEL);
WRITELN(DOSTRICH); WRITELN(' '); WRITELN(M);
WRITELN(' ');
WRITELN(STRICH, STRICH, STRICH, STRICH, STRICH);
WRITE('HINWEIS: VARIABLE VOM TYP INTEGER QUERF');
WRITELN('EN WERTE');
WRITE('AUS DEM INTERVALL VON ', -MAXINT, ' BIS ');
WRITELN(MAXINT);
WRITELN('ANNEHMEN!');
WRITELN(STRICH, STRICH, STRICH, STRICH, STRICH);
END. (* INVENTUR *)
```

```
PRO T O K O L L D E R
I N V E N T U R I M L A G E R X Y Z:
-----
170 SENKKOPFSCHRAUBEN VOM TYP 123
225 SENKKOPFSCHRAUBEN VOM TYP 125
73130 SENKKOPFSCHRAUBEN VOM TYP 135
33130 SENKKOPFSCHRAUBEN VOM TYP 150
-----
106655 SENKKOPFSCHRAUBEN
=====
MAGDEBURG, DEN
-----
HINWEIS: VARIABLE VOM TYP INTEGER QUERFEN WERTE
AUS DEM INTERVALL VON -2147483647 BIS 2147483647
ANNEHMEN!
-----
```

Kugel

FUER EINEN GEGEBENEN GANZZAHLIGEN WERT DES RADIUS SIND DIE OBERFLAECHE UND DAS VOLUMEN EINER KUGEL K ZU BERECHNEN UND DIE ERGEBNISSE ZU DRUCKEN.

```
PROGRAM KUGEL(OUTPUT);
(* ARBEIT M. VARIABLEN VON VERSCHIEDENEN TYPEN *)
(* AUSDRUCK, STANDARD TYPEN *)
CONST
PI=3,1415926;
VAR
RADIUS: INTEGER; (* VARIABLE V. GANZZAHLIGEM TYP *)
OFLAECHE, VOLUMEN: REAL;
BEGIN
WRITELN('K U G E L'); WRITELN('GEGEBEN IST DER');
RADIUS:=10; (* ANWEISUNG, DIE VAR, WERT ZUWEIST *)
WRITELN('GANZZAHLIGE RADIUS=', RADIUS, ', ');
OFLAECHE:=4*PI*RADIUS*RADIUS; WRITELN(' ');
WRITELN('BERECHNUNGSERGEBNISSE FUER KUGEL K :');
WRITELN('OBERFLAECHE=', OFLAECHE, ', ');
VOLUMEN:=4*PI*RADIUS*RADIUS*RADIUS/3;
WRITELN('KUG. VOLUMEN=', VOLUMEN, ', ');
END. (* KUGEL *)
```

```
K U G E L
GEGEBEN IST DER
GANZZAHLIGE RADIUS= 10.
BERECHNUNGSERGEBNISSE FUER KUGEL K :
OBERFLAECHE= 1.25663E+03,
KUG. VOLUMEN= 4.18678E+03.
```

Berechnungen mehrerer Kreise

FUER MEHRERE KREISE SIND DER DURCHMESSER, DER UMFANG UND DER FLAECHEINHALT ZU BERECHNEN UND ZU DRUCKEN. DIE RADII SIND GEGEBEN!

```
PROGRAM KREIS02(OUTPUT);
(* ARBEIT MIT VARIABLEN VOM TYP REAL *)
(* VARIABLENDEKLARATION, ANWEISUNG, AUSDRUCK *)
CONST
(* DARF NUR EINMAL AUFTRETEN! *)
```

```
BERECHNUNGEN VON GROESSEN DES KREISES
1. KREIS, GEGEBEN IST
RADIUS = 0,99999E+01.
```

```
PI=3,1415926;
VAR
RADIUS, DURCHMESSER, UMFANG, FLAECHE: REAL;
BEGIN
WRITELN('BERECHNUNGEN VON GROESSEN DES KREISES');
WRITELN(' '); WRITELN('1. KREIS, GEGEBEN IST ');
RADIUS:=10; WRITELN(' RADIUS =', RADIUS, ', ');
UMFANG:=2*PI*RADIUS;
FLAECHE:=PI*RADIUS*RADIUS;
DURCHMESSER:=2*RADIUS;
WRITELN(' UMFANG =', UMFANG);
WRITELN(' FLAECHE =', FLAECHE);
WRITE(' DURCHMESSER =', DURCHMESSER, ' WURDEN BER');
WRITELN('ECHNET, ');
WRITELN(' '); WRITELN('2. KREIS, GEGEBEN IST ');
RADIUS:=100; WRITELN(' RADIUS =', RADIUS, ', ');
FLAECHE:=PI*RADIUS*RADIUS;
UMFANG:=2*PI*RADIUS;
WRITELN(' UMFANG =', UMFANG);
WRITELN(' FLAECHE =', FLAECHE); WRITELN('.....');
WRITELN('BEACHTEN SIE DIE ZAHLNDARSTELLUNGEN!');
END. (* KREIS02 *)
```

```
UMFANG = 6.28318E+01
FLAECHE= 3.14159E+02
DURCHMESSER= 2,00000E+01 WURDEN BERECHNET.
2. KREIS, GEGEBEN IST
RADIUS = 0,99999E+02.
UMFANG = 6.28318E+02
FLAECHE= 3.14159E+04
*****
BEACHTEN SIE DIE ZAHLNDARSTELLUNGEN!
```

Plan-Abrechnung

FUER 5 BRIGADEN IST DIE ERFUELLUNG DER 5 SCHICHTEN EINER WOCHEN ABZURECHNEN. DAZU IST DIE LAUFENDE NUMMER DER WOCHEN UND VON JEDER BRIGADE IHR MAXIMAL SIEBENSTELLIGER NAME UND DIE 5 PROZENTZAHLEN DER PLANERFUELLUNG EINZULESEN. NEBEN DIESEN DATEN IST DIE DURCHSCHNITTLICHE PLANERFUELLUNG ZU DRUCKEN. DIE DATEN EINER BRIGADE BEFINDEN SICH JEWEILS AUF EINER LOCKKARTE. ALLE PROZENTANGABEN SIND GANZZAHLIG!

```
PROGRAM PLANIST(INPUT, OUTPUT);
(* EIN-/AUSGABE VON TEXT UND ZAHLN *)
(* PROGRAMMPARAMETER, READ, WRITE, CHAR, INTEGER *)
CONST
N='BRIGADE'; LEER=' ';
STRICH='-----';
VAR
A, B, C, D, E, F, G: CHAR; (* ZUM EINLESEN DER NAMEN *)
SMO, SDI, SMI, SDO, SFR, (* PLANERFUELLUNG *) WNR,
DMO, DDI, DMI, DDO, DFR (* DURCHSCHNITTE *) : INTEGER;
BEGIN
READLN(WNR); (* EINLESEN DER NUMMER DER WOCHEN *)
WRITELN('PLAN-ABRECHNUNG DER', WNR, '3, ' WOCHE: ');
WRITELN(STRICH, STRICH); WRITELN(' ');
WRITELN('MEISTERBEREICH: 301');
WRITELN('MON:29, 'DIE:4, 'MIT:4, 'DON:4, 'FRE:');
WRITELN(STRICH, STRICH);
READLN(A, B, C, D, E, F, G, SMO, SDI, SMI, SDO, SFR);
WRITELN('LEER, N, A, B, C, D, E, F, G, SMO:4, SDI:4, SMI:4,
SDO:4, SFR:4, (SMO+SDI+SMI+SDO+SFR) DIV 5:5);
DMO:=SMO; DDI:=SDI; DMI:=SMI; DDO:=SDO; DFR:=SFR;
READLN(A, B, C, D, E, F, G, SMO, SDI, SMI, SDO, SFR);
WRITELN('LEER, N, A, B, C, D, E, F, G, SMO:4, SDI:4, SMI:4,
SDO:4, SFR:4, (SMO+SDI+SMI+SDO+SFR) DIV 5:5);
DMO:=DMO+SMO; DDI:=DDI+SDI; DMI:=DMI+SMI;
DDO:=DDO+SDO; DFR:=DFR+SFR;
READLN(A, B, C, D, E, F, G, SMO, SDI, SMI, SDO, SFR);
WRITELN('LEER, N, A, B, C, D, E, F, G, SMO:4, SDI:4, SMI:4,
SDO:4, SFR:4, (SMO+SDI+SMI+SDO+SFR) DIV 5:5);
DMO:=DMO+SMO; DDI:=DDI+SDI; DMI:=DMI+SMI;
DDO:=DDO+SDO; DFR:=DFR+SFR;
READLN(A, B, C, D, E, F, G, SMO, SDI, SMI, SDO, SFR);
WRITELN('LEER, N, A, B, C, D, E, F, G, SMO:4, SDI:4, SMI:4,
SDO:4, SFR:4, (SMO+SDI+SMI+SDO+SFR) DIV 5:5);
DMO:=DMO+SMO; DDI:=DDI+SDI; DMI:=DMI+SMI;
DDO:=DDO+SDO; DFR:=DFR+SFR;
WRITELN(STRICH, STRICH);
WRITELN('DURCHSCHN. PLANERFUELLUNG');
WRITE('IN PROZENT ');
WRITE(DMO DIV 5:4, DDI DIV 5:4, DMI DIV 5:4);
WRITELN(DDO DIV 5:4, DFR DIV 5:4);
WRITELN(STRICH, STRICH, STRICH, STRICH, STRICH);
WRITELN('DIE DURCHSCHNITTSWERTE DER BRIGADEN ');
WRITELN('UND SCHICHTEN SIND GERUNDET !');
END. (* PLANIST *)
```

```
PLAN-ABRECHNUNG DER 17. WOCHEN:
-----
MEISTERBEREICH: 301
MON DIE MIT DON FRE
-----
BRIGADE BRUESER 104 111 109 101 100 105
BRIGADE SIMMEL 100 104 114 114 100 106
BRIGADE HERTEL 103 104 105 106 100 103
BRIGADE SCHUART 101 115 107 104 100 105
BRIGADE TESZMER 100 118 109 110 100 107
-----
DURCHSCHN. PLANERFUELLUNG
IN PROZENT 101 110 108 107 100
-----
DIE DURCHSCHNITTSWERTE DER BRIGADEN
UND SCHICHTEN SIND GERUNDET !
```



```

READLN(A,B); P:=A*B; S:=S+P; I:=I+1;
WRITELN(I:2,A:18,B:15,P:15);
WRITELN(STR:48,'-');
WRITELN('SKALARES PRODUKT=':37,S:13);
WRITELN(STR,STR,STR,STR,STR);
END. (* PRODUKTE *)

```

Netz eines Tetraeders

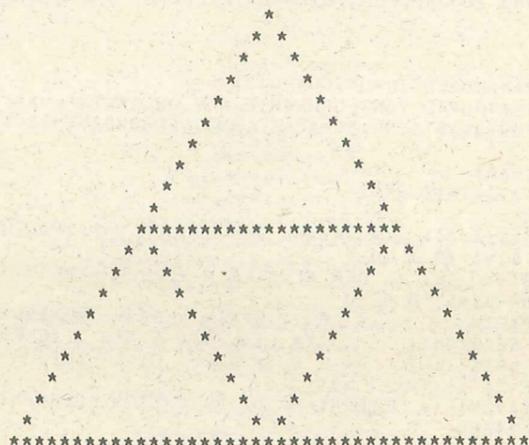
DAS NETZ EINES TETRAEDERS IST ZU ERZEUGEN UND ZU DRUCKEN. FUER DIE DARSTELLUNG DIESER ABWICKLUNG SIND "*" ZEICHEN ZU VERWENDEN.

```

PROGRAM TETRANET(OUTPUT);
(* GRAFIK: NETZ EINES TETRAEDERS *)
(* FOR=ANWEISUNG,FELDBREITE,AUSDRUCK *)
CONST
N=26; (* POSITION DER SPITZE DES DREIECKS *)
VAR
I:INTEGER;
BEGIN
WRITELN('NETZ EINES TETRAEDERS ');
WRITELN(' ':50);
WRITELN('*':N); (* SPITZE DES DREIECKS *)
FOR I:=1 TO 9 DO WRITELN('*':N-I,'*':2*I);
WRITE('*':N-10);
FOR I:=1 TO 20 DO WRITE('*');
WRITELN; (* OBERES TEILDREIECK *)
FOR I:=1 TO 9 DO
WRITELN('*':N-10-I,'*':2*I,'*':20-2*I,'*':2*I);
WRITE('*':N-20);
FOR I:=1 TO 40 DO WRITE('*');
WRITELN (* UNTERE TEILDREIECKE *)
END. (* TETRANET *)

```

NETZ EINES TETRAEDERS



Einführung von Vergleichsoperatoren

DIE NUTZUNGSMOEGELICHKEITEN VON VERGLEICHOPERATOREN. FUER DEN VERGLEICH VON VARIABLEN UND/ODER KONSTANTEN SIND ZU DEMONSTRIEREN.

```

PROGRAM GLEICH(INPUT,OUTPUT);
(* VERGLEICHE ZWISCHEN VARIABLEN; KONSTANTEN *)
(* BOOLESCHER=AUSDRUCK,VERGLEICHOPERATOREN *)
CONST
STR='-----'; STRICH='-----';
K1='KONDENSATOR 1'; K2='KONDENSATOR 2';
K3='KAPAZITAET'; IN1='INDUKTIVITAET';
VAR
C1,C2,C3,C4:CHAR;
A1,A2,A3,A4:ALFA;
I1,I2,I3,I4:INTEGER;
R1,R2,R3,R4:REAL;
BEGIN
WRITELN('EINFUEHRUNG VON VERGLEICHOPERATOREN:');
WRITELN('=<> <= >= > <':32);
WRITELN(STRICH,STRICH,STR);
WRITELN('AUSDRUECKE BESTEHEN AUS OPERATOREN UND',
'OPERANDEN. ');
WRITELN('AUSDRUECKE LIEFERN EINEN WERT. BEI VER',
'GLEICHEN IST');
WRITELN('DER RESULTATTYP "BOOLEAN". DER OPERATU',
'R <> BEDEU-');

```

```

EINFUEHRUNG VON VERGLEICHOPERATOREN:
= <> <= >= > <
-----
AUSDRUECKE BESTEHEN AUS OPERATOREN UND OPERANDEN.
AUSDRUECKE LIEFERN EINEN WERT. BEI VERGLEICHEN IST
DER RESULTATTYP "BOOLEAN". DER OPERATOR <> BEDEU-
TET "UNGLEICH", <= "KLEINER ODER GLEICH" UND >= BE
DEUTET "GRUESZER ODER GLEICH".
-----
WERTZUWEISUNGEN:
I1= 100 I2= 200 I3= 300
R1= 150,4998 R2= 200,4997 R3= 3,0075E+02
VERGLEICHE:
I1=I2: FALSE. I2<I3: TRUE
R1>R2: FALSE. I1<R2: TRUE
R2>R3: FALSE. K2<I3: TRUE
-----
TEST AUF LEXIKOGRAPHISCHE ORDNUNG V. ZEICHENKETTEN
VERGLEICHE:
KONDENSATOR 1 <= KONDENSATOR 2: TRUE
KONDENSATOR 1 <> KAPAZITAET: TRUE
NUR GLEICH LANGE ZEICHENKETTEN I

```

```

WRITELN('TET "UNGLEICH", <= "KLEINER ODER GLEIC',
'H" UND >= BE');
WRITELN('DEUTET "GRUESZER ODER GLEICH".');
WRITELN(STRICH,STRICH,STR);
READLN(C1,C2,C3,C4); READLN(I1,I2,I3,I4);
READLN(R1,R2,R3,R4);
A1:='BRUESER'; A2:='SIMMEL'; A3:='HERTEL';
A4:='TESZMER';
WRITELN('WERTZUWEISUNGEN ':45);
WRITELN('I1=',I1:4,' I2=',I2:4,' I3=',I3:4);
WRITELN('R1=',R1:8:4,' R2=',R2:8:4,' R3=',R3:12);
WRITELN('VERGLEICHE:');
WRITELN('I1=I2:',I1=I2:6,' I2<I3:',I2<I3:6);
WRITELN('R1>R2:',R1>R2:6,' I1<R2:',I1<R2:6);
WRITELN('R2>R3:',R2>R3:6,' R2<I3:',R2<I3:6);
WRITELN(STRICH:35);
WRITELN('TEST AUF LEXIKOGRAPHISCHE ORDNUNG V. Z',
'EICHENKETTEN');
WRITELN('VERGLEICHE:');
WRITELN(K1,' <= ',K2,' :',K1<=K2:6);
WRITELN(K1,' <> ',K3,' :',K1<>K3:6);
WRITELN('NUR GLEICH LANGE ZEICHENKETTEN I');
WRITELN('WERTZUWEISUNGEN ':45);
WRITELN('A1=',A1); WRITELN('A2=',A2,' C1=',C1);
WRITELN('A3=',A3); WRITELN('A4=',A4,' C2=',C2);
WRITELN('VERGLEICHE:');
WRITELN('A1<A2:',A1<A2:6,' C1<C2:',C1<C2);
WRITELN('A2<A3:',A2<A3:6,' C1>C2:',C1>C2);
WRITELN('A3<A4:',A3<A4:6,' C1=C2:',C1=C2);
WRITELN('A2>A4:',A2>A4:6);
WRITELN(STRICH,STRICH,STR);
END. ($ GLEICH *)

```

WERTZUWEISUNGEN:

A1= BRUESER			
A2= SIMMEL	C1=A		
A3= HERTEL			
A4= TESZMER	C2=B		
VERGLEICHE:			
A1<A2=	TRUE	C1<C2=	TRUE
A2<A3=	FALSE	C1>C2=	FALSE
A3<A4=	TRUE	C1=C2=	FALSE
A2>A4=	FALSE		

Skalares Produkt

ZUR BERECHNUNG DES SKALAREN PRODUKTES ZWEIER VEKTOREN SIND IHRE DIMENSION UND JEWEILS ENTSPRECHENDE KOMPONENTEN, DIE AUF LOCHKARTEN PAARWEISE BEREITGESTELLT SIND, EINZULESEN UND ZU PROTOKOLLIEREN.

```

PROGRAM SKALPRO(INPUT,OUTPUT);
(* SKALARES PRODUKT ZWEIER VEKTOREN *)
(* FOR=ANWEISUNG,VERBUNDANWEISUNG,READLN,WRITE *)
VAR
I,DIM:INTEGER;
A,B,SKP:REAL;
BEGIN
WRITELN('SKALARES PRODUKT ZWEIER ');
WRITELN('VEKTOREN ':); WRITELN(' ':50);
READLN(DIM); WRITELN('DIMENSION=',DIM:3,' ');
WRITELN(' ');
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
WRITELN('INDEX','A[I]':16,'B[I]':20);
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
SKP:=0; (* ANFANGSWERT AN SUMMATIONS VARIABLE *)
FOR I:=1 TO DIM DO
BEGIN (* DER VERBUNDANWEISUNG *)
READLN(A,B); (* EINLESEN EINES WERTEPAARES *)
WRITELN(I:5,A:20,B:20);
SKP:=SKP+A*B; (* SCHRITTHALTENDE SUMMATION *)
END; (* OF FOR I *)
WRITELN('-----':45);
WRITELN('SKALARES PRODUKT=',SKP:13);
WRITELN('=====':30);
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* SKALPRO *)

```

SKALARES PRODUKT ZWEIER VEKTOREN:

DIMENSION= 15.

INDEX	A [I]	B [I]
1	-1,00000E+00	-1,00000E+00
2	2,00000E+00	5,00000E-01
3	3,00000E+00	3,33333E-01
4	-4,00000E+00	-2,50000E-01
5	5,00000E+00	2,00000E-01
6	6,00000E+00	1,66667E-01
7	7,00000E+00	1,42857E-01
8	8,00000E+00	1,25000E-01
9	9,00000E+00	1,11111E-01
10	0,99999E+01	1,00000E-01
11	1,10000E+01	9,09089E-02
12	1,20000E+01	8,33332E-02
13	1,30000E+01	7,69230E-02
14	1,40000E+01	7,14284E-02
15	1,50000E+01	6,66665E-02

SKALARES PRODUKT= 1,50000E+01
=====

Lage eines Punktes

BEI DER POSITIONIERUNG EINES WERKZEUGES BZW ROBOTERARMES IST ZU PRUEFEN, OB EIN PUNKT P MIT DEN KOORDINATEN X,Y IM INNERN EINES RECHTWINKLIGEN DREIECKES LIEGT, VON DEN BEIDEN KATHETEN SOLL P MINDESTENS C LAENGENEINHEITEN ENTFERNT SEIN, DIE MASZAHLEN BEIDER KATHETEN UND DIE WERTE FUER X UND Y SIND EINZULESEN UND ZU DRUCKEN, DIE KATHETEN LIEGEN AUF DER X- BZW Y-ACHSE.

```

PROGRAM LAGE(INPUT,OUTPUT);
(* ENTHALTENSEINS-TEST *)
(* BOOLEAN,IF=ANWEISUNG,VERGLEICHOPERATOREN *)
VAR
L1,L2,L3,L:BOOLEAN;
X,Y,A,B,C:REAL;
BEGIN

```

TEST OB EIN PUNKT IN EINEM DREIECK LIEGT:

GEGEBEN SIND DIE BEIDEN LAENGEN DER KATHETEN DES DREIECKS:
A= 10,000
B= 15,745
UND DER ABSTAND C= 0,200.

```

WRITE('TEST OB EIN PUNKT IN EINEM DREIECK LIEGT',
      ' '); WRITELN; WRITELN(' ',50);
READLN(A,B); (* KATHETENLAENGEN DES DREIECKS *)
READLN(C); (* WERT DES ABSTANDES C EINLESEN *)
WRITELN('GEGEBEN SIND DIE BEIDEN LAENGEN DER ');
WRITELN('KATHETEN DES DREIECKS:');
WRITELN('A=',A:8:3); WRITELN('B=',B:8:3);
WRITELN('UND DER ABSTAND C=',C:6:3,' ');
WRITELN(' ',25); WRITELN('ZU TESTEN SIND:');
READLN(X,Y); (* KOORDINATEN DES PUNKTES P *)
WRITELN('KOORDINATEN X,Y DES PUNKTES P(X,Y): ');
WRITELN('X=',X:10,X:7:3); WRITELN('Y=',Y:10,Y:7:3);
L1:=Y>=C; (* VERGLEICH OB ORDINATE >= C IST *)
L2:=X>=C; (* VERGLEICH OB ABSZISSE >= C IST *)
L3:=Y<(-B/A)*X+B; (* VERGLEICH Y<HYPOTHENUSE *)
L:=L1 AND L2 AND L3;
WRITELN(' '); WRITELN('TEST - ERGEBNIS: ');
WRITELN(' ',50); WRITE(' ',22);
IF L THEN WRITELN('P ERFUELLT DIE BEDINGUNGEN !')
ELSE WRITELN('BEDINGUNGEN NICHT ERFUELLT!');
END. (* LAGE *)

```

```

ZU TESTEN SIND:
KOORDINATEN X,Y DES PUNKTES P(X,Y):
      X= 1,750
      Y= 10,224
*
T E S T - E R G E B N I S :
      P ERFUELLT DIE BEDINGUNGEN !

```

Manipulation mit Worten

UNTER VERWENDUNG VON VARIABLEN DES TYP ALFA SIND MANIPULATIONEN MIT ZEICHENFOLGEN DER KONSTANTEN LAENGE ACHT AUSZUFUEHREN UND DEREN ERGEBNISSE ZU DRUCKEN.

```

PROGRAM ALFATYP(OUTPUT);
(* GRAFIK MIT VARIABLEN VOM TYP ALFA *)
(* FOR=ANWEISUNG,ALFA,ERGIBTANWEISUNG *)
CONST
STR='-----'; A='ABCDEFGH'; ST='-----';
L='          '; (* 16 LEERZEICHEN *)
VAR
A1,A2,A3,A4:ALFA; (* VARIABLE VOM TYP ALFA *)
I:INTEGER;
BEGIN
WRITELN('DEMONSTRATION VON VARIABLEN VOM TYP "ALFA" ');
WRITELN('ALFA:');
A1:=TESTWORT; A2:=A; A3:=ST;
A4:=A1; (* WERTZUWEISUNGEN AN VARIABLE *)
WRITELN(' ');
WRITELN(A3,A4,A3,A3,A3:2,A4,A3);
WRITELN(' ',50);
FOR I:=8 DOWNTO 1 DO
WRITELN(L,A1:I,' ',18-2*I,A1:I);
(* BEACHTEN SIE DIESE FORM DER FOR=ANWEISUNG! *)
FOR I:=2 TO 8 DO
WRITELN(L,A1:I,' ',18-2*I,A1:I);
WRITELN(' ',50);
FOR I:=1 TO 7 DO
WRITELN(L,' ',8-I,A2:I,' ',A2:I);
WRITELN(A2,A2,A2,A2:2,A2,A2);
WRITELN(STR,STR,STR,STR,STR); WRITELN(' ');
WRITE('VARIABLEN VOM TYP ALFA DUERFEN NUR ZEICHENKETTEN');
WRITELN('HENKETTEN');
WRITE('DER KONSTANTEN LAENGE 8 ZUGEWIESEN WERDEN');
WRITELN('ENI');
WRITELN(STR,STR,STR,STR,STR);
END. (* ALFATYP *)

```

```

DEMONSTRATION VON VARIABLEN VOM TYP "ALFA" :
-----TESTWORT-----TESTWORT-----
TESTWORT TESTWORT
TESTWOR TESTWOR
TESTWO TESTWO
TESTW TESTW
TEST TEST
TES TES
TE TE
T T
TE TE
TES TES
TEST TEST
TESTW TESTW
TESTWO TESTWO
TESTWOR TESTWOR
TESTWORT TESTWORT
A A
AB AB
ABC ABC
ABCD ABCD
ABCDE ABCDE
ABCDEF ABCDEF
ABCDEFG ABCDEF
ABCDEFGHABCDEFGHABCDEFGHABCDEFGH
VARIABLEN VOM TYP ALFA DUERFEN NUR ZEICHENKETTEN
DER KONSTANTEN LAENGE 8 ZUGEWIESEN WERDEN!

```

Signifikante Stellen

DIE "AUSWIRKUNGEN" DER VERWENDUNG VON BEZEICHNERN, DIE IN DEN ERSTEN ACHT ZEICHEN UEBEREINSTIMMEN SIND ZU DEMONSTRIEREN.

```

PROGRAM NAMEN(OUTPUT);
(* SIGNIFIKANTE STELLEN EINES BEZEICHNERS *)
(* VARIABLE,WRITELN *)
CONST
LANGNAME1='***';
VAR
NAMESLANG:INTEGER;
BEGIN
NAMESLANG:=10101010;
NAMESLANG:=12345678;
WRITE('SIGNIFIKANTE STELLEN');

```

```

WRITELN('INEN, PROGRAMMEN UND VON');
WRITE('DATEN SOWIE ANZEIGEFELDERN IN DATENSAET');
WRITELN('ZEN. ');
WRITE('LANGNAME1=:25,LANGNAME1:5,NAMELANG1:');
WRITE('NAMELANG1:8,NAMELANG=:NAMELANG:8, ');
WRITELN('LANGNAMEBEISPIEL=:LANGNAMEBEISPIEL:5);
WRITELN('NAMELANG2=:25,NAMELANG2:8);
WRITELN('ACHTUNG:');
WRITE('BEZEICHNER, DIE IN DEN ERSTEN ACHT ZE');
WRITE('ICHEN UEBEREINSTIMMEN WERDEN VOM COMPIL');
WRITELN('ER NICHT UNTERSCHIEDEN!');

```

```

WRITELN('EINES BEZEICHNERS. ');
WRITE('BEZEICHNER DIENEN ALS NAMEN VON KONSTANTEN');
WRITE('TEN, TYPEN, VARIABLEN, PARAMETERN, ROUTINEN');

```

```

WRITE('BEZEICHNER MUESSEN VON ALLEN WORTSYMBOL ');
WRITELN('EN VERSCHIEDEN SEIN!');
END. (* NAMEN *)

```

```

SIGNIFIKANTE STELLEN EINES BEZEICHNERS.
BEZEICHNER DIENEN ALS NAMEN VON KONSTANTEN, TYPEN, VARIABLEN, PARAMETERN, ROUTINEN, PROGRAMMEN UND VON
DATEN SOWIE ANZEIGEFELDERN IN DATENSAETZEN.
LANGNAME1='***' NAMELANG1=12345678 NAMELANG=12345678 LANGNAMEBEISPIEL='***'
NAMELANG2=12345678

```

ACHTUNG: BEZEICHNER, DIE IN DEN ERSTEN ACHT ZEICHEN UEBEREINSTIMMEN WERDEN VOM COMPILER NICHT UNTERSCHIEDEN! BEZEICHNER MUESSEN VON ALLEN WORTSYMBOLEN VERSCHIEDEN SEIN!

Qualitätskontrolle

BEI EINER AUTOMATISIERTEN QUALITAETSKONTROLLE FUER PROFILSTAHL WIRD DER ZUSCHNITT UEBERPRUEFT. EINZULESEN SIND DIE WERTE FUER DEN SOLLWERT DER LAENGE, DIE ANZAHL ZU PRUEFENDER OBJEKTE, DIE PROBENUMMER UND DIE ANFALLENDEN MESZDATEN. ZU ERMITTELN SIND DER GROESZTE UND KLEINSTE WERT UND DER DURCHSCHNITTSWERT. EIN ENTSPRECHENDES PROTOKOLL IST ZU DRUCKEN.

```

PROGRAM QUALITAE(INPUT,OUTPUT);
(* MESZDATENVERARBEITUNG Z. QUALITAETSPRUEFUNG *)
(* FOR=ANWEISUNG,IF=ANWEISUNG,READ,WRITE *)
CONST
U='UEBERLAENGE=';
VAR
B:BOOLEAN;
I,N,PROBE,ANZAHL:INTEGER;
X,S,MAX,MIN,LAENGE:REAL;
BEGIN
WRITELN('VEB KOMBINAT EDELSTAHL');
WRITELN('ABTEILUNG QUALITAETSKONTROLLE');
WRITELN('---:50);
WRITELN('PROTOKOLL DER');
WRITELN('STICHPROBENENTNAHME');
READLN(PROBE); WRITELN('---:50);
WRITELN('NR,DER STICHPROBE: ',46,PROBE:4);
READLN(LAENGE); (* VORGEGEBENER SOLLWERT *)
WRITELN('SOLLWERT DER LAENGE=',LAENGE:10:3);
READLN(ANZAHL); (* ZU PRUEFENDER OBJEKTE *)
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
WRITELN('NR, 'IST-LAENGE', 'EINSTUFUNG':22);
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
READLN(X); (* 1. MESZWERT EINES OBJEKTES *)
MAX:=X; MIN:=X; S:=X; I:=1; (* ANFANGSWERTE *)
B:=X < LAENGE;
IF B THEN WRITELN(I:3,X:11:3,'AUSSCHUSZ!':36)
ELSE WRITELN(I:3,X:11:3,U,LAENGE-X:6:3);
FOR I:=2 TO ANZAHL DO
BEGIN
READLN(X);
B:=X < LAENGE;
S:=S+X;
IF MAX < X THEN MAX:=X;
IF MIN > X THEN MIN:=X;
IF B THEN WRITELN(I:3,X:11:3,'AUSSCHUSZ!':36)
ELSE WRITELN(I:3,X:11:3,U,LAENGE-X:6:3);
END; (* OF FOR I *)
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
WRITELN('KLEINSTE IST-LAENGE=',MIN:10:3);
WRITELN('GROESZTE IST-LAENGE=',MAX:10:3);
WRITELN('DURCHSCHNITTS-WERT =',S/ANZAHL:10:3);
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* QUALITAE *)

```

```

VEB KOMBINAT EDELSTAHL
ABTEILUNG QUALITAETSKONTROLLE
-----
PROTOKOLL DER
STICHPROBENENTNAHME
-----
NR,DER STICHPROBE: 138
SOLLWERT DER LAENGE= 4,750
-----
NR. IST-LAENGE EINSTUFUNG
-----
1 4,760 UEBERLAENGE=-0,010
2 4,752 UEBERLAENGE=-0,002
3 4,760 UEBERLAENGE=-0,010
4 4,780 UEBERLAENGE=-0,030
5 4,740 AUSSCHUSZ!
6 4,752 UEBERLAENGE=-0,002
7 4,752 UEBERLAENGE=-0,002
8 4,811 UEBERLAENGE=-0,061
9 4,560 AUSSCHUSZ!
10 4,760 UEBERLAENGE=-0,010
11 4,750 UEBERLAENGE= 0
12 4,811 UEBERLAENGE=-0,061
13 4,880 UEBERLAENGE=-0,130
14 4,760 UEBERLAENGE=-0,010
15 4,760 UEBERLAENGE=-0,010
16 4,880 UEBERLAENGE=-0,130
17 4,880 UEBERLAENGE=-0,130
18 4,883 UEBERLAENGE=-0,133
19 4,883 UEBERLAENGE=-0,133
20 4,760 UEBERLAENGE=-0,010
21 4,733 AUSSCHUSZ!
22 4,722 AUSSCHUSZ!
23 4,733 AUSSCHUSZ!
24 4,775 UEBERLAENGE=-0,026
25 4,791 UEBERLAENGE=-0,041
26 4,883 UEBERLAENGE=-0,133
-----
KLEINSTE IST-LAENGE= 4,560
GROESZTE IST-LAENGE= 4,883
DURCHSCHNITTS-WERT = 4,781
-----

```

Nullteiler

AN BEISPIELEN IST ZU ZEIGEN, DASZ BEIM RECHNEN MIT REELLEN GROESZEN SUGENANNT "NULLTEILER" AUFTRETEN KOENNEN.

```

PROGRAM NULLTEIL(OUTPUT);
(* NACHWEIS VON NULLTEILERN *)
(* KONSTANTE,WRITELN *)
CONST
K1=2,0E-40; K2=1,5E-39; K3=5,0E+12;
BEGIN
WRITELN('EXISTENZ VON NULLTEILERN:');
WRITELN('-----');

```

```

EXISTENZ VON NULLTEILERN:
-----
DIE FAKTOREN
K1= 2,0000E-40
K2= 1,5000E-39
SIND BEIDE UNGLEICH NULL.
IHR PRODUKT IST 0!
K3*K2*K1= 1,50000E-66

```

```

WRITELN('DIE FAKTOREN'); WRITELN('K1=',K1:12);
WRITELN('K2=',K2:12);
WRITELN('SINUS BEI DER UNGLEICHUNG ULL');
WRITELN('IHR PRODUKT IST',K1*K2:13,' ');
WRITELN('K3*K2*K1=',K3*K2*K1);
WRITELN('K1*K3*K2=',K1*K3*K2);
WRITELN('K1*K2*K3=',K1*K2*K3);
WRITELN('VORSICHT BEI DER DIVISION, DAMIT DER NENNER NICHT
'ENNER NICHT ');
WRITELN('GLEICHUNG ULL IST ');
END. (* NULLTEIL *)

```

```

K1*K3*K2= 1.50000E-66
K1*K2*K3= 0
VORSICHT BEI DER DIVISION, DAMIT DER NENNER NICHT
GLEICHUNG ULL IST !

```

Masse-Berechnung eines Rohteils

AUS EINER METALLPLATTE KONSTANTER DICHTER RO UND KONSTANTER DICKE D IST EIN DREIECK MIT DEN SEITENLAENGEN A,B,C AUSZUSCHNEIDEN. DIE MASSE DIESES ROHTEILS IST ZU BERECHNEN. DIE WERTE FUER DIE VARIABLEN A,B,C,D UND RO SIND EINZULESEN. DABEI IST ZU PRUEFEN, OB FUER DIE EINGELESENEN WERTE DIE SEITEN A,B,C TATSACHLICH EIN DREIECK AUFSPANNEN. DIE BERECHNUNG IST IN EINER GEFIEGNEIEN FORM ZU PROTOKOLLIEREN.

```

PROGRAM MASSE(INPUT,OUTPUT);
(* MASSE EINES ROHTEILS *)
(* BOOLEAN-TYP, IF-ANWEISUNG *)
CONST
ST=-----; S=''; M=' [M]';
L='';
VAR
A,B,C,D,RO,VOL,FL,HUM,RAD,MAS:REAL;
F,L1,L2,L3:BOOLEAN;
BEGIN
WRITELN('MASSE - BERECHNUNG EINES ROHTEILS');
WRITELN('EINGABE WERTE:');
READLN(A,B,C,D,RO);
WRITELN('GEBEN SIND DIE WERTE FOLGENDER GROESZEN:');
WRITELN(L,'SEITE A =',A:7:3,M);
WRITELN(L,'SEITE B =',B:7:3,M);
WRITELN(L,'SEITE C =',C:7:3,M);
WRITELN(L,'DICKE D =',D:7:3,M);
WRITELN(L,'DICHTER RO =',RO:7:3,' [KG/(M**M**M)]');
L1:=A < (B+C);
L2:=B < (A+C);
L3:=C < (A+B);
F:=L1 AND L2 AND L3;
IF F THEN BEGIN
HUM:=(A+B+C)/2; (* HALBER UMFANG *)
RAD:=HUM*(HUM-A)*(HUM-B)*(HUM-C);
FL:=SQRT(RAD); (* FORMEL VON HERON *)
VOL:=FL*D;
MAS:=VOL*RO;
WRITELN('BERECHNET WURDEN DIE WERTE',
'FOLGENDER GROESZEN:');
WRITELN(L,'FLAECHE =',FL:7:3,' [M**M]');
WRITELN(L,'VOLUMEN =',VOL:7:3,' [M**M**M]');
WRITELN(L,'MASSE =',MAS:7:3,' [KG]');
WRITELN(L,'=====');
END
ELSE WRITELN('ES WURDEN KEINE WERTE BERECHNET,
'ET, DA DATENFEHLER!');
WRITELN(S,S:49);
WRITELN(ST,ST:10);
END. (* MASSE *)

```

```

MASSE - BERECHNUNG EINES ROHTEILS :
-----
1
GEBEN SIND DIE WERTE FOLGENDER GROESZEN:
SEITE A = 3,000 [M]
SEITE B = 4,000 [M]
SEITE C = 5,000 [M]
DICKE D = 0,115 [M]
DICHTER RO = 7,850 [KG/(M**M**M)]:
BERECHNET WURDEN DIE WERTE FOLGENDER GROESZEN:
FLAECHE = 6,000 [M**M]
VOLUMEN = 0,690 [M**M**M]
MASSE = 5,416 [KG]
=====
1

```

Zeichentabelle

DIE MENGE DER 256 VERFUEGBAREN ZEICHEN VOM TYP CHAR IST ZU DRUCKEN. DIESE ZEICHEN SIND IN EINER MATRIX ANZUORDNEN, DIE ZUGLEICH DEN ZUSAMMENHANG ZWISCHEN ZEICHEN UND SEINEM WERT ERKENNEN LAESZT. EINE ENTSPRECHENDE BESCHRIFTUNG DIESER TABELLE IST VORZUNEHMEN.

```

PROGRAM ZEICHENTAB (OUTPUT);
(* ZEICHENTABELLE *)
(* FOR-ANWEISUNG, CHAR-TYP, STANDARDFUNKTIONEN *)
VAR
I,J: INTEGER;
BEGIN
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
WRITE('TABELLE DER 256 BUCHSTABEN, ZIFFERN');
WRITELN('UND SPEZIALSYMBOLE DES COMPILERS. ');
END; (* FOR *)
WRITE(' '); WRITELN;
FOR J:=1 TO 57 DO WRITE(' '); WRITELN;
WRITELN; (* BEGRENZUNGSSTRICH WIRD GEDRUCKT. *)
WRITE(' '); WRITELN;
FOR J:=0 TO 15 DO WRITE(J:3);
WRITELN(' '); (* TABELLENFUSZ WIRD GEDRUCKT. *)
WRITELN('HINWEISE: ');

```

```

END; (* FOR *)
WRITE(' '); WRITELN;
FOR J:=1 TO 57 DO WRITE(' '); WRITELN;
WRITELN; (* BEGRENZUNGSSTRICH WIRD GEDRUCKT. *)
WRITE(' '); WRITELN;
FOR J:=0 TO 15 DO WRITE(J:3);
WRITELN(' '); (* TABELLENFUSZ WIRD GEDRUCKT. *)
WRITELN('HINWEISE: ');

```

```

WRITE('DIE AUF DEM BENUTZTEN DRUCKER NICHT VERFUEGBAREN
ZEICHEN WERDEN DURCH LEERZEICHEN (BLANKS) DARGESTELLT. ');
WRITELN(' ');
WRITE(' '); WRITELN;
FOR J:=0 TO 15 DO WRITE(J:3);
WRITELN(' '); (* TABELLENKOPF WIRD GEDRUCKT. *)
WRITE(' '); WRITELN;
FOR J:=1 TO 57 DO WRITE(' '); WRITELN;
WRITELN; (* BEGRENZUNGSSTRICH WIRD GEDRUCKT. *)
FOR I:=0 TO 15 DO
BEGIN WRITE(I:26,' ');
FOR J:=0 TO 15 DO WRITE(CHR(16*I+J):3);
WRITELN(' ',I:2);

```

```

WRITE('DIE KLEINBUCHSTABEN WERDEN ALS GROSZBUCHSTABEN
GEDRUCKT. ');
WRITELN('HSTABEN GEDRUCKT. ');
WRITE('DIE 26 BUCHSTABEN DES ALPHABETES FOLGEN NICHT
DICHT AUF EINANDER, ZWISCHEN I UND J BZW R UND S SIND
SPEZIALSYMBOLE EINGEFUEGT. ');
WRITE('DER WERT EINES ZEICHENS ERGIBT SICH AUS "16*
ZEILENNUMMER + SPALTENNUMMER". IHN LIEFERT DIE
FUNKTION I:=ORD(ZEICHEN). DIE FUNKTION CHR(I) STELLT
DAS ZEICHEN DAR, DESSEN WERT DEM VON I GLEICH IST. ');
END. (* ZEICHENTAB *)

```

TABELLE DER 256 BUCHSTABEN, ZIFFERN UND SPEZIALSYMBOLE DES COMPILERS. DIE AUF DEM BENUTZTEN DRUCKER NICHT VERFUEGBAREN ZEICHEN WERDEN DURCH LEERZEICHEN (BLANKS) DARGESTELLT.

	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	!
0	!																	!
1	!																	!
2	!																	!
3	!																	!
4	!																	!
5	!																	!
6	!																	!
7	!																	!
8	!																	!
9	!																	!
10	!																	!
11	!																	!
12	!																	!
13	!																	!
14	!																	!
15	!																	!
	!	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	!

HINWEISE:
DIE KLEINBUCHSTABEN WERDEN ALS GROSZBUCHSTABEN GEDRUCKT.
DIE 26 BUCHSTABEN DES ALPHABETES FOLGEN NICHT DICHT AUF EINANDER. ZWISCHEN I UND J BZW R UND S SIND SPEZIALSYMBOLE EINGEFUEGT.
DER WERT EINES ZEICHENS ERGIBT SICH AUS "16*ZEILENNUMMER + SPALTENNUMMER". IHN LIEFERT DIE FUNKTION I:=ORD(ZEICHEN). DIE FUNKTION CHR(I) STELLT DAS ZEICHEN DAR, DESSEN WERT DEM VON I GLEICH IST.

Druck reeller Größen

POTENZEN VON 2 MIT NEGATIVEN EXPONENTEN SIND SOWOHL IN FESTPUNKT- ALS AUCH IN GLEITPUNKT-DARSTELLUNG ZU TABELLIEREN.

```

PROGRAM ZWEIHOCH(OUTPUT);
(* FEST-/GLEITPUNKTDARSTELLUNG VON ZAHLEN *)
(* FOR-ANWEISUNG, VERBUNDANWEISUNG, WRITE *)
CONST
N=30;
VAR
I: INTEGER;
POTENZ: REAL;
BEGIN
WRITELN(' ');
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
WRITELN('POTENZEN VON 2 MIT NEGATIVEN EXPONENTEN');
WRITE(' '); WRITELN;
FOR I:=1 TO 34 DO WRITE(' '); WRITELN;
WRITELN('EXPO- GLEITPUNKT- FESTPUNKT- ');
WRITELN('NENT DARSTELLUNG DARSTELLUNG ');
WRITE(' '); WRITELN;
FOR I:=1 TO 34 DO WRITE(' '); WRITELN;
POTENZ:=1;
FOR I:=1 TO N DO
BEGIN
POTENZ:=POTENZ/2;
WRITELN(-I:14,POTENZ:13,POTENZ:16:13);
END;
WRITELN(' ');
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;

```

```

-----
POTENZEN VON 2 MIT NEGATIVEN EXPONENTEN
-----
EXPO- GLEITPUNKT- FESTPUNKT-
NENT DARSTELLUNG DARSTELLUNG
-----
-1 5.00000E-01 0.500000000000000
-2 2.50000E-01 0.250000000000000
-3 1.25000E-01 0.125000000000000
-4 6.25000E-02 0.062500000000000
-5 3.12500E-02 0.0312500059604
-6 1.56250E-02 0.015625000000000
-7 7.81250E-03 0.007812500000000
-8 3.90625E-03 0.0039062505960
-9 1.95312E-03 0.001953125000000
-10 9.76562E-04 0.000976562500000
-11 4.88281E-04 0.0004882813096
-12 2.44141E-04 0.0002441406250
-13 1.22070E-04 0.0001220703721
-14 6.10352E-05 0.0000610351860
-15 3.05176E-05 0.0000305175900
-16 1.52588E-05 0.0000152587950
-17 7.62939E-06 0.0000076293945
-18 3.81470E-06 0.0000038146972
-19 1.90735E-06 0.0000019073486

```

```

WRITELN('A C H T U N G !');
WRITELN('INTERNE DARSTELLUNG DER ZAHLEN BEACHTE',
      'N!');
END. (* ZWEIHOCH *)

```

```

-20 9,53674E-07 0,0000009536743
-21 4,76837E-07 0,0000004768373
-22 2,38419E-07 0,0000002384186
-23 1,19209E-07 0,0000001192093
-24 5,96047E-08 0,000000596047
-25 2,98023E-08 0,000000298023
-26 1,49012E-08 0,000000149012
-27 7,45059E-09 0,000000745056
-28 3,72529E-09 0,00000037253
-29 1,86264E-09 0,00000018626
-30 9,31323E-10 0,00000009313

```

A C H T U N G !
 INTERNE DARSTELLUNG DER ZAHLEN BEACHTEN!

Schachbrett-Muster

ES IST EIN SCHACHBRETT-ARTIGES MUSTER ZU ERZEUGEN UND MIT ENTSPRECHENDER BESCHRIFTUNG DER ZEILEN UND SPALTEN AUSZUGEBEN.

```

PROGRAM SCHABRET(OUTPUT);
(* SCHACHBRETT *)
(* FOR=ANWEISUNG,IF=ANWEISUNG,CHAR=TYP *)
VAR
  I:INTEGER;
  Z:CHAR;
BEGIN
  WRITELN('S C H A C H B R E T T':35);
  WRITELN('MUSTER':28); WRITELN('!':50);
  WRITELN('!');
  WRITE('!':17); (* LAUFVARIABLEN=TYP BEACHTEN ! *)
  FOR Z:='A' TO 'H' DO WRITE(Z:2); WRITELN;
  FOR I:=8 DOWNTO 1 DO
    BEGIN
      WRITE(I:17); (* WENN I UNGERADE,IST ODD=TRUE *)
      IF ODD(I) THEN WRITELN(' # # # # # ',I:1)
        ELSE WRITELN(' # # # # # ',I:1)
    END;
    WRITE('!':17);
  FOR Z:='A' TO 'H' DO WRITE(Z:2); WRITELN;
  END. (* SCHABRET *)

```

S C H A C H B R E T T
 -MUSTER

	A	B	C	D	E	F	G	H
8	#	#	#	#	#	#	#	8
7	#	#	#	#	#	#	#	7
6	#	#	#	#	#	#	#	6
5	#	#	#	#	#	#	#	5
4	#	#	#	#	#	#	#	4
3	#	#	#	#	#	#	#	3
2	#	#	#	#	#	#	#	2
1	#	#	#	#	#	#	#	1
	A	B	C	D	E	F	G	H

Nachbildung einer Schaltung

DIE ARBEITSWEISE EINER HALBADDERER-SCHALTUNG ZUR SUMMATION ZWEIER EINSTELLIGER D U A L - ZAHLEN SA,SB IST DURCH EIN PROGRAMM NACHZUBILDEN. DIESE SCHALTUNG BESITZT ZWEI EINGAENGE, DENEN DIE BEIDEN SUMMANDEN UND ZWEI AUSGAENGE, DENEN DIE SPALTENSUMME SU UND DER UEBERTRAG UE ZUGEORDNET SIND.

```

PROGRAM HADDER(INPUT,OUTPUT);
(* HALB=ADDIERER-SCHALTUNG *)
(* BOOLEAN=TYP,IF=ANWEISUNG,FOR=ANWEISUNG *)
CONST
  SU='SUMMAND'; ST='*****';
  UT='UEBERTRAG'; PF='----->'; S='!';
  SP='SPALTENSUMME';
VAR
  A1,B1,SS,UB:BOOLEAN;
  SA,SB,SM,UE,I:INTEGER;
BEGIN
  WRITELN('HALBADDERER-SCHALTUNG':36);
  WRITELN(ST,ST,ST:10);
  WRITELN(S,S:49);
  WRITELN(ST:35);
  WRITELN(S:16,S:19);
  WRITELN(S:16,S:19);
  WRITELN(SU:12,'1 *','!':20,SP);
  WRITELN(PF:15,'*','!':19,PF);
  WRITELN(S:16,S:19);
  WRITELN(S:16,'HALBADDERER':15,S:4);
  WRITELN(S:16,S:19);
  WRITELN(SU:12,'2 *','!':20,UT);
  WRITELN(PF:15,'*','!':19,PF);
  WRITELN(S:16,S:19);
  WRITELN(S:16,S:19);
  WRITELN(ST:35);
  WRITELN(S,S:49);
  WRITELN(ST,ST,ST:10);

```

HALBADDERER-SCHALTUNG

```

*****
*
*
SUMMAND 1 * * SPALTENSUMME
----->* *----->
*
*
* HALBADDERER *
*
*
SUMMAND 2 * * UEBERTRAG
----->* *----->
*
*
*****
*
*
BESCHREIBUNG DER SCHALTUNGS-ZUSTAENDE
*
# SUMMAND 2 SUMMAND 1 UEBERTRAG SPALTENSUMME
*****
0 0 0 0 0
1 0 1 0 1
2 1 0 0 1
3 1 1 1 0
*****

```

```

WRITELN(S,S:49);
WRITE(' BESCHREIBUNG DER SCHALTUNGS-ZUSTAENDE');
WRITELN; WRITELN(S,S:49);
WRITE(' #',SU:11,'2 ',SU,'1 ',UT,' ',SP);
WRITELN; WRITELN(' ',ST,ST,ST:6);
FOR I:=0 TO 3 DO
  BEGIN
    READLN(SA,SB);
    IF SA=1 THEN A1:=TRUE ELSE A1:=FALSE;
    IF SB=1 THEN B1:=TRUE ELSE B1:=FALSE;
    SS:=(A1 AND NOT B1) OR (NOT A1 AND B1);
    UB:=A1 AND B1;
    IF SS THEN SM:=1 ELSE SM:=0;
    IF UB THEN UE:=1 ELSE UE:=0;
    WRITELN(I:3,SB:7,SA:10,UE:10,SM:10);
  END;
  WRITELN(S,S:49);
  WRITELN(ST,ST,ST:10);
END. (* HADDER *)

```

Boolesche Größen

DIE NUTZUNG VON LAUFVARIABLEN VOM TYP BOOLEAN IST ZU ZEIGEN. DABEI BESCHRAENKE MAN SICH AUF 2 VARIABLE.

```

PROGRAM BOOLE(OUTPUT);
(* LOGISCHE OPERATOREN *)
(* FOR=ANWEISUNG,BOOLEAN=TYP *)
VAR
  P,Q:BOOLEAN;
BEGIN
  WRITELN('FOR=ANWEISUNGEN MIT LAUFVARIABLEN VOM ',
    'TYP BOOLEAN'); WRITELN('---','!':47);
  FOR P:=FALSE TO TRUE DO
    FOR Q:=FALSE TO TRUE DO
      BEGIN
        WRITELN(P:17,' UND',Q:6,' IST',P AND Q:6);
        WRITELN(P:17,' ODER',Q:6,' IST',P OR Q:6);
      END; WRITELN('-----','!':38);
    END. (* BOOLE *)

```

FOR=ANWEISUNGEN MIT LAUFVARIABLEN VOM TYP BOOLEAN

```

---
FALSE UND FALSE IST FALSE
FALSE ODER FALSE IST FALSE
FALSE UND TRUE IST FALSE
FALSE ODER TRUE IST TRUE
TRUE UND FALSE IST FALSE
TRUE ODER FALSE IST TRUE
TRUE UND TRUE IST TRUE
TRUE ODER TRUE IST TRUE
-----

```

Werbung

EIN MUSTER FUER EIN PLAKAT ZUR WERBUNG FUER EIN "STUDIUM DER INFORMATIK" IST ZU DRUCKEN! DAS WORT "STUDIUM" SOLL SICH DABEI WIEDERHOLEN!

```

PROGRAM WERBUNG(OUTPUT);
(* WERBUNG FUER EIN STUDIUM *)
(* FELD,FOR=ANWEISUNG *)
VAR
  I,J:INTEGER;
  FELD:ARRAY[1..10] OF CHAR;
BEGIN
  FELD[1]:='I'; FELD[2]:='N'; FELD[3]:='F';
  FELD[4]:='O'; FELD[5]:='R'; FELD[6]:='M';
  FELD[7]:='A'; FELD[8]:='T'; FELD[9]:='FELD[1]';
  FELD[10]:='K';
  FOR I:=1 TO 10 DO
    BEGIN
      WRITE('!':22=I);
      FOR J:=1 TO I DO WRITE(FELD[J]);
      WRITELN(' - S T U D I U M ');
    END;
  FELD[1]:='-'; FELD[2]:='S'; FELD[3]:='T';
  FELD[4]:='U'; FELD[5]:='D'; FELD[6]:='I';
  FELD[7]:='U'; FELD[8]:='M'; FELD[9]:='!';
  FOR I:=1 TO 9 DO WRITELN(FELD[I]:22);
  WRITELN('STUDIENMOEGLICHKEITEN AN DER');
  WRITELN('UNIVERSITAET ROSTOCK');
  WRITELN('INGENIEURHOCHSCHULE DRESDEN');
  WRITELN('TECHNISCHEN UNIVERSITAET DRESDEN');
  WRITELN('TECHNISCHEN HOCHSCHULE MAGDEBURG');
  WRITELN('TECHNISCHEN HOCHSCHULE KARL-MARX-STADT');
  WRITELN; WRITELN(' ');
  FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
  WRITELN('WERBUNG FUER INFORMATIK');
  END. (* WERBUNG *)

```

```

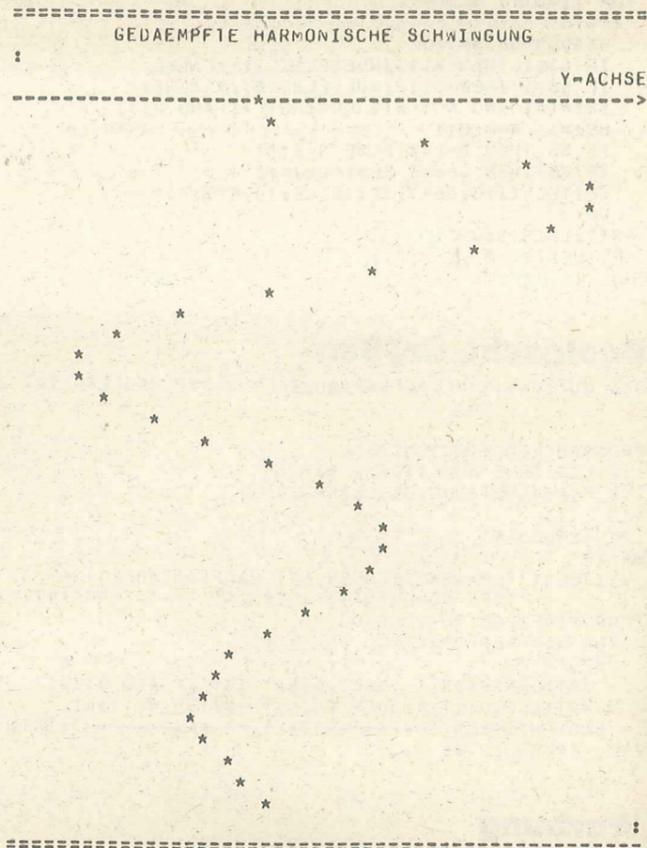
I - S T U D I U M
IN - S T U D I U M
INF - S T U D I U M
INFO - S T U D I U M
INFORM - S T U D I U M
INFORMA - S T U D I U M
INFORMAT - S T U D I U M
INFORMATI - S T U D I U M
INFORMATIK - S T U D I U M
-
S
T
U
D
I
U
M
!
STUDIENMOEGLICHKEITEN AN DER
.
UNIVERSITAET ROSTOCK
INGENIEURHOCHSCHULE DRESDEN
TECHNISCHEN UNIVERSITAET DRESDEN
TECHNISCHEN HOCHSCHULE MAGDEBURG
TECHNISCHEN HOCHSCHULE KARL-MARX-STADT
.
#####
WERBUNG FUER INFORMATIK

```

Darstellung einer Schwingung

EINE GEDAEMPFTHE HARMONISCHE SCHWINGUNG KANN DURCH EINE ZEITFUNKTION $Y=A*EXP(ALFA*T)*SIN(OMEGA*T+FI)$, WO BEI $ALFA < 0$, BESCHRIEBEN WERDEN. GESUCHT IST EINE GRAPHISCHE DARSTELLUNG VON Y UEBER DER T-ACHSE.

```
PROGRAM SCHWING(OUTPUT);
(* GRAFIK: GEDAEMPFTHE HARMONISCHE SCHWINGUNG *)
(* REPEAT=ANWEISUNG, FOR=ANWEISUNG, STANDARDFKT. *)
CONST
  A=1; (* AMPLITUDEN=FAKTOR *)
  ALFA=-1;
  FI=0; (* ANFANGSPHASE *)
  OMEGA=6,28318; (* SCHWINGUNGS-FREQUENZ *)
  D=0,0625; (* =1/16 :16 STUETZWERTE IN T...T+1 *)
  S=32; (* DISKRETE WERTE IN Y...Y+1 *)
  H=20; (* ABSTAND DER T-ACHSE VOM L. RAND *)
  ANZ=32; (* ANZAHL DER STUETZSTELLEN *)
VAR
  I,ORD:INTEGER;
  Y,T:REAL;
BEGIN
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  WRITELN('GEDAEMPFTHE HARMONISCHE SCHWINGUNG:41 ');
  WRITELN(' '); WRITELN('Y-ACHSE:50');
  FOR I:=1 TO H-1 DO WRITE(' '); WRITELN(' ');
  FOR I:=H+1 TO 49 DO WRITE(' '); WRITELN(' ');
  FOR I:=0 TO ANZ DO
    BEGIN
      T:=D*I;
      Y:=A*EXP(ALFA*T)*SIN(OMEGA*T+FI);
      ORD:=ROUND(S*Y)+H; (* ROUND:STANDARDFUNKTION *)
      REPEAT (* BEGINN DER ZYKLUS-ANWEISUNG *)
        WRITE(' '); (* "AUFFUELLEN" MIT BLANKS *)
        ORD:=ORD-1;
      UNTIL ORD=0; (* ENDE DER REPEAT-ANWEISUNG *)
      WRITELN(' ');
    END; (* OF FOR *)
  WRITELN(' ');
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* SCHWING *)
```



Bildauswertung

IN EINEM AUTOMATISCHEN BILDAUSWERTUNGSSYSTEM WIRD IN DER ZEITEINHEIT EINE JEWEILS UNTERSCHIEDLICHE ANZAHL VON ELEMENTAROBJEKTEN ERFASST. DIESE AUSWERTUNG BRICHT AB, SOBALD DIESE ANZAHL GLEICH NULL IST. ES WIRD DAS MAXIMUM UND DAS VON NULL VERSCHIEDENE MINIMUM DER REGISTRIERTEN WERTE UND IHRE ANZAHL GESUCHT.

```
PROGRAM BILDAUS(INPUT,OUTPUT);
(* MAX,MIN EINER FOLGE *)
(* WHILE=ANWEISUNG, IF=ANWEISUNG, FOR=ANWEISUNG *)
VAR
  BEDINGUNG:BOOLEAN;
  ANZAHL,ZAEHLER,MAX,MIN,I,NR:INTEGER;
BEGIN
  WRITELN('P R O T O K O L L D E R');
  READLN(NR);
  WRITELN('AUSWERTUNG DER ELEMENTAROBJEKTE NR: ',
    NR); WRITELN(' '); WRITELN(' ');
  WRITELN('ZEIT ANZAHL MAXIMUM MINIMUM:39');
  FOR I:=1 TO 39 DO IF I<12 THEN WRITE(' ')
    ELSE WRITE(' ');
  WRITELN;
  MIN:=MAXINT; (* ANFANGSWERTZUWEISUNGEN *)
  MAX:=MININT;
  ZAEHLER:=0;
  BEDINGUNG:=TRUE;
  WHILE BEDINGUNG DO
    BEGIN
      READ(ANZAHL);
      IF ANZAHL=0
        THEN BEDINGUNG:=FALSE
        ELSE BEGIN
          ZAEHLER:=ZAEHLER+1;
          IF ANZAHL<MIN THEN MIN:=ANZAHL;

```

PROTOKOLL DER
AUSWERTUNG DER ELEMENTAROBJEKTE NR: 103

ZEIT	ANZAHL	MAXIMUM	MINIMUM
1	255	255	255
2	289	289	255
3	255	289	255
4	265	289	255
5	255	289	255
6	256	289	255
7	301	301	255
8	256	301	255
9	333	333	255
10	277	333	255
11	265	333	255
12	265	333	255
13	265	333	255
14	265	333	255
15	256	333	255
16	257	333	255
17	256	333	255
18	257	333	255
19	257	333	255
20	265	333	255
21	257	333	255

```
IF ANZAHL>MAX THEN MAX:=ANZAHL;
WRITELN(ZAEHLER:15,ANZAHL:8,MAX:8,
  MIN:8);
END (* IF *)
END; (* WHILE *)
READLN; (* ABSCHLUSS DER EINGABE *) WRITELN(' ');
WRITELN('ES WURDEN',ZAEHLER:3,' ZEITEINHEITEN ',
  'AUSGEWERTET. ');
WRITELN('MAXIMUM=',MAX:4,' ELEMENTAROBJEKTE/ZE ');
WRITELN('MINIMUM=',MIN:4,' ELEMENTAROBJEKTE/ZE ');
END. (* BILDAUS *)
```

22	265	333	255
23	265	333	255
24	265	333	255
25	149	333	149
26	211	333	149
27	199	333	149

ES WURDEN 27 ZEITEINHEITEN AUSGEWERTET.
MAXIMUM= 333 ELEMENTAROBJEKTE/ZE
MINIMUM= 149 ELEMENTAROBJEKTE/ZE

Alphabet

ALLE ZEICHEN DES ALPHABETES SIND SO SPALTENVERSETZT ZU DRUCKEN, DASS SIE AUF EINER "DIAGONALEN" ANGEORDNET SIND. DIE LEXIKOGRAPHISCHE ANORDNUNG IST ZU GEWAHRLEISTEN.

```
PROGRAM WHILEDEM(OUTPUT);
(* ALPHABET *)
(* WHILE=ANWEISUNG,STANDARDFUNKTION *)
VAR
  CH:CHAR;
  I:INTEGER;
BEGIN
  WRITELN('DEMONSTRATION DER W H I L E - ANWEISU',
    'NG MIT VA=');
  WRITELN('RIABLEN VOM TYP CHAR');
  I:=38; (* POSITIONIERUNG DES ERSTEN BUCHSTABENS *)
  CH:='A';
  WHILE CH<='I' DO BEGIN
    WRITELN(' ',CH,CH,CH,CH,CH );
    CH:=SUCC(CH); (* SUCC IST EINE
  STANDARDFUNKT. *)
    I:=I-1;
  END; (* OF WHILE *)
  CH:='J';
  WHILE CH<='R' DO BEGIN
    WRITELN(' ',CH,CH,CH,CH,CH );
    CH:=SUCC(CH);
    I:=I-1;
  END;
  CH:='S';
  WHILE CH<='Z' DO BEGIN
    WRITELN(' ',CH,CH,CH,CH,CH );
    CH:=SUCC(CH);
    I:=I-1;
  END;
  I:=1; WRITELN(' ');
  WHILE I<=50 DO BEGIN
    WRITE(' ');
    I:=I+1;
  END; WRITELN; (* DOPPELSTRICH *)
  WRITELN('SIEHE AUCH PROGRAMM "ZEICHTAB":41');
END. (* WHILEDEM *)
```

DEMONSTRATION DER W H I L E - ANWEISUNG MIT VA-RIABLEN VOM TYP CHAR

```
AAAA
BBBB
CCCC
DDDD
EEEE
FFFF
GGGG
HHHH
IIII
JJJJ
KKKK
LLLL
MMMM
NNNN
O000
PPPP
QQQQ
RRRR
SSSS
TTTT
UUUU
VVVV
wwww
XXXX
YYYY
ZZZZ
```

SIEHE AUCH PROGRAMM "ZEICHTAB":1

Elementare Funktionen

ELEMENTARE FUNKTIONEN MIT GANZZAHLIGEM ARGUMENT-WIE DIE SUMME NATUERLICHER ZAHLEN, IHR PRODUKT, DIE 2.-POTENZ UND DIE 2.- UND 4.-WURZEL - SIND ZU BERECHNEN UND ZU TABELLIEREN.

```
PROGRAM FUNKTION(OUTPUT);
(* TABELLIERUNG VERSCHIEDENER FUNKTIONEN *)
(* STANDARDFUNKTION,FUNKTION, FOR=ANWEISUNG *)
CONST
  STR='-----'; ANZ=22;
VAR
  I,SI,II,SII:INTEGER;
  PI:REAL;
BEGIN
  WRITELN('ELEMENTARE FUNKTIONEN MIT GANZZAHLIG',
    'EM ARGUMENT');
  WRITELN(STR,STR,STR,STR,STR);
  WRITELN(' I SI II SII SQRT(I) 4.WURZEL ',
    ' PI');
  WRITELN(STR,STR,STR,STR,STR);
  SI:=0; PI:=1; SII:=0; (* ANFANGSWERTZUWEISUNGEN *)
  FOR I:=1 TO ANZ DO
    BEGIN
      SI:=SI+I; (* SUMMIERUNG DER WERTE VON I *)

```

I	SI	II	SII	SQRT(I)	4.WURZEL	PI
1	1	1	1	0,9999996	0,9999996	1,000000E+00
2	3	4	5	1,4142125	1,1892062	2,000000E+00
3	6	9	14	1,7320501	1,3160735	6,000000E+00
4	10	16	30	1,9999992	1,4142125	2,400000E+01
5	15	25	55	2,2360664	1,4953482	1,200000E+02
6	21	36	91	2,4494886	1,5650838	7,200000E+02
7	28	49	140	2,6457500	1,6265755	5,040000E+03
8	36	64	204	2,8284257	1,6817921	4,032000E+04
9	45	81	285	2,9999989	1,7320501	3,628800E+05
10	55	100	385	3,1622755	1,7782783	3,628800E+06
11	66	121	506	3,3166229	1,8211591	3,991680E+07
12	78	144	650	3,4641003	1,8612086	4,790010E+08
13	91	169	819	3,6055499	1,8988287	6,227020E+09
14	105	196	1015	3,7416553	1,9343358	8,717820E+10
15	120	225	1240	3,8729810	1,9679892	1,307670E+12

```

PI:=PI*I; (* FAKULTAETSBERECHNUNG *)
II:=SQR(I); (* STANDARDFUNKTION "QUADRAT" *)
SII:=SII+II; (* SUMMIERUNG DER QUADRATE *)
WRITELN(I:2,SII:4,II:4,SII:5,SQR(I):10:7,
        SQR(SQR(I)):10:7,PI);
END;
WRITELN(STR,STR,STR,STR,STR);
WRITELN('LEGENDE:':24);
WRITELN('SUMME VON 1 BIS I          SII:50);
WRITELN('QUADRATE VON I           II:50);
WRITELN('SUMME DER QUADRATE VON 1 BIS I SII:50);
WRITELN('QUADRATWURZEL AUS I      SQR(I):50);
WRITELN('VIERTE WURZEL AUS I     4.WURZEL:50);
WRITELN('PRODUKT VON 1 BIS I     PI:50);
WRITELN(STR,STR,STR,STR,STR);
END. (* FUNKTION *)

```

16	136	256	1496	3,9999985	1,9999992	2,09227E+13
17	153	289	1785	4,1231042	2,0305424	3,55687E+14
18	171	324	2109	4,2426389	2,0597654	6,40236E+15
19	190	361	2470	4,3588972	2,0877969	1,21645E+17
20	210	400	2870	4,4721335	2,1147412	2,43289E+18
21	231	441	3311	4,5825737	2,1406942	5,10908E+19
22	253	484	3795	4,6904134	2,1657359	1,12400E+21

```

LEGENDE:
SUMME VON 1 BIS I          SII
QUADRATE VON I           II
SUMME DER QUADRATE VON 1 BIS I SII
QUADRATWURZEL AUS I      SQR(I)
VIERTE WURZEL AUS I     4.WURZEL
PRODUKT VON 1 BIS I     PI

```

Zweierpotenzen

DIE POTENZEN VON ZWEI MIT NEGATIVEN EXPONENTEN SIND IN DEZIMALER DARSTELLUNG ALS TABELLE ZU DRUCKEN. ES IST EINE FORM DER BERECHNUNG "O H N E RUNDUNGSFEHLER" ZU WAELLEN.

```

PROGRAM POTENZEN(OUTPUT);
(* NEGATIVE POTENZEN VON 2 OHNE RUNDUNGSFEHLER *)
(* TYPDEFINITION,TEILBEREICHSTYPEN,FELD *)
CONST
N=45; (* MAXIMALER EXPONENT *)
TYPE
ZIFFER=0..9; (* TEILBEREICHSTYP *)
VAR
I,J,R:INTEGER;
Z:ARRAY[1..N] OF ZIFFER;
BEGIN
WRITELN('POTENZEN VON 2 IN DEZIMALER',
        'DARSTELLUNG');
WRITELN('MIT NEGATIVEN EXPONENTEN UND OHNE RU',
        'NDUNGSFEHLER');
WRITELN(' ');
WRITELN('POTENZ', '(=-1)*EXPONENT:44);
FOR I:=1 TO 50 DO WRITE(' ');
WRITELN; (* BEGRENZUNGS=STRICH *)
FOR J:=1 TO N DO
BEGIN
WRITE('0,');
R:=0;
FOR I:=1 TO J-1 DO
BEGIN
R:=10*R+Z[I];
Z[I]:=R DIV 2;
R:=R-2*Z[I];
WRITE(CHR(Z[I]+ORD('0')));
END; (* FOR I *)
Z[J]:=5; (* LETZTE ZIFFER DER ZWEIER=POTENZ *)
WRITELN(' ',J:48-J)
END (* FOR J *)
END. (* POTENZEN *)

```

POTENZEN VON 2 IN DEZIMALER DARSTELLUNG
MIT NEGATIVEN EXPONENTEN UND OHNE RUNDUNGSFEHLER

POTENZ	(-1)*EXPONENT
0,5	1
0,25	2
0,125	3
0,0625	4
0,03125	5
0,015625	6
0,0078125	7
0,00390625	8
0,001953125	9
0,0009765625	10
0,00048828125	11
0,000244140625	12
0,0001220703125	13
0,00006103515625	14
0,000030517578125	15
0,0000152587890625	16
0,00000762939453125	17
0,000003814697265625	18
0,0000019073486328125	19
0,00000095367431640625	20
0,000000476837158203125	21
0,0000002384185791015625	22
0,00000011920928955078125	23
0,000000059604644775390625	24
0,0000000298023223876953125	25
0,00000001490116119384765625	26
0,000000007450580596923828125	27
0,0000000037252902984619140625	28
0,00000000186264514923095703125	29
0,000000000931322574615478515625	30
0,0000000004656612873077392578125	31
0,00000000023283064365386962890625	32
0,000000000116415321826934814453125	33
0,0000000000582076609134674072265625	34
0,00000000002910383045673370361328125	35
0,000000000014551915228366851806640625	36
0,0000000000072759576141834259033203125	37
0,00000000000363797880709171295166015625	38
0,000000000001818989403545856475830078125	39
0,0000000000009094947017729282379150390625	40
0,00000000000045474735088646411895751953125	41
0,000000000000227373675443232059478759765625	42
0,0000000000001136868377216160297393798828125	43
0,00000000000005684341886080801486968994140625	44
0,000000000000028421709430404007434844970703125	45

Sollwertabweichung

BEI EINEM EXPERIMENT SOLL DIE TEMPERATUR DES ZU UNTERSUCHENDEN MEDIUMS EINEN VORGEGEBENEN SOLLWERT TSOL BESITZEN. ZUR KONTROLLE WERDEN IN AEQUIDISTANTEN ZEITPUNKTEN DIE TEMPERATURWERTE T[I] IN GRAD CELSIUS REGISTRIERT. NACH N ZEIT-EINHEITEN SIND FUER JEDEN MESZWERT SEINE ABWEICHUNG AT SOL VOM SOLLWERT, SEINE ABWEICHUNG AMIT VOM MITTELWERT UEBER DIE N WERTE T[I] UND DIE STANDARDABWEICHUNG ZU BERECHNEN.

```

PROGRAM SOLLWERT(INPUT,OUTPUT);
(* MESZDATEN=AUSWERTUNG *)
(* FELDTYPEN,FOR=ANWEISUNG *)
CONST
N=32;
ST='=====';
VAR
I,NR:INTEGER;
TSOL,ATSOL,AMIT,SAMIT,MIT:REAL;
T:ARRAY[1..N] OF REAL; (* VARIABLE VOM FELDTYP *)
BEGIN
WRITELN(ST,ST,ST:10);
WRITELN('P R O T O K O L L DER TEMPERATURUEBER',
        'WACHUNG');
READLN(NR); (* LIES NUMMER DES EXPERIMENTES! *)
WRITELN('EXPERIMENT-NR.:46,NR:4);
WRITELN(ST,ST,ST:10);
READLN(TSOL); (* LIES TEMPERATUR=SOLLWERT *)
WRITELN('TEMPERATUR=SOLLWERT =',TSOL:6:2);
WRITELN('ANZAHL DER MESZWERTE=',N:3);
WRITELN(ST,ST,ST:10);
MIT:=0;
FOR I:=1 TO N DO
BEGIN
READ(T[I]); (* EINLESEN DER MESZDATEN *)
MIT:=MIT+T[I];
END; (* FOR I *)
READLN;
MIT:=MIT/N; (* ABSCHLUSS MIT-BERECHNUNG *)
SAMIT:=0;
WRITELN(' I ':18,' T[I] ':7,' T[I]-TSOL ':10,
        ' T[I]-MIT ':9);
WRITELN(ST:36,ST:8);
FOR I:=1 TO N DO
BEGIN
AMIT:=1[I]-MIT;
ATSOL:=T[I]-TSOL;
WRITELN(I:18,T[I]:7:2,ATSOL:10:2,AMIT:9:2);
SAMIT:=SAMIT+SQR(T[I]-MIT);
END; (* FOR I *)
WRITELN(ST,ST,ST:10);
SAMIT:=SQR(SAMIT/(N-1));
WRITELN('MITTELWERT=',MIT:6:2);
WRITELN('STANDARDABWEICHUNG=',SAMIT:5:2);
WRITELN(ST,ST,ST:10);
END. (* SOLLWERT *)

```

=====

P R O T O K O L L DER TEMPERATURUEBERWACHUNG

EXPERIMENT-NR.: 133

=====

TEMPERATUR=SOLLWERT = 80,00

ANZAHL DER MESZWERTE= 32

=====

I	T[I]	T[I]-TSOL	T[I]-MIT
1	80,50	0,50	0,45
2	79,95	-0,05	-0,10
3	79,50	-0,50	-0,55
4	80,10	0,10	0,05
5	80,30	0,30	0,25
6	80,00	-0,00	-0,05
7	80,90	0,90	0,85
8	78,90	-1,10	-1,15
9	79,10	-0,90	-0,95
10	79,70	-0,30	-0,35
11	81,00	1,00	0,95
12	81,10	1,10	1,05
13	81,20	1,20	1,15
14	80,10	0,10	0,05
15	80,20	0,20	0,15
16	80,10	0,10	0,05
17	79,90	-0,10	-0,15
18	79,85	-0,15	-0,20
19	79,90	-0,10	-0,15
20	80,10	0,10	0,05
21	80,40	0,40	0,35
22	80,30	0,30	0,25
23	80,20	0,20	0,15
24	78,80	-1,20	-1,25
25	79,40	-0,60	-0,65
26	79,70	-0,30	-0,35
27	79,90	-0,10	-0,15
28	80,00	-0,00	-0,05
29	80,10	0,10	0,05
30	80,10	0,10	0,05
31	80,10	0,10	0,05
32	80,10	0,10	0,05

=====

MITTELWERT= 80,05

STANDARDABWEICHUNG= 0,55

=====

Werkzeugverschleißnorm

ZUR UEBERPRUEFUNG DER ERZEUGNISBEZOGENEN VERSCHLEISSNORMEN VON WERKZEUGEN IST EIN "HISTOGRAMM" DER LEBENSDAUER DERSELBEN AUFZUSTELLEN UND ZU DRUCKEN.

```

PROGRAM WERKZEUG(INPUT,OUTPUT);
(* LEBENSDAUER=HISTOGRAMM VON WERKZEUGEN *)
(* FELD,TEILBEREICHSTYPEN,FOR=ANWEISUNG *)
CONST
DAUER=25; (* MAXIMALE LEBENSDAUER IN STUNDEN *)
VAR
INDEX:0..DAUER; (* TEILBEREICHSTYP *)
HISTO:ARRAY[0..DAUER] OF INTEGER; (* FELDTYP *)
WERT,I,ANZAHL:INTEGER;
BEGIN
WRITELN('H I S T O G R A M M DER LEBENSDAUER: ',
        '"WERKZEUGE"');
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
FOR INDEX:=0 TO DAUER DO HISTO[INDEX]:=0;
WRITELN(' ':50);
READLN(ANZAHL); (* ANZAHL DER WERKZEUGE *)
FOR I:=1 TO ANZAHL DO
BEGIN
READ(WERT); (* BLANKS,ZEILENENDE ALS TRENNZ. *)
IF WERT IN [0..DAUER] (* VERGLEICHSUPERATOR *)
(* [0..DAUER] BESCHREIBT DIE MENGE ALLER WERTE
IM ABGESCHLOSSENEN INTERVALL VON 0..DAUER. *)
THEN BEGIN
INDEX:=WERT;
HISTO[INDEX]:=HISTO[INDEX]+1
END
ELSE WRITELN('WERT',WERT:4,' LIEGT AUSZERHAL',
        'B DES BEREICHES!')
END; (* FOR I *)

```

H I S T O G R A M M DER LEBENSDAUER: "WERKZEUGE".

=====

WERT 27 LIEGT AUSZERHALB DES BEREICHES!

WERT 26 LIEGT AUSZERHALB DES BEREICHES!

=====

LD	ANZAHL
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1
8	1
9	1
10	1
11	1
12	0
13	1
14	1
15	1
16	1
17	2
18	3
19	6
20	19

```

WRITELN('=:50);
READLN; (* ABSCHLUSS DER EINGABE DER WERTE *)
WRITELN('LD ANZAHL:29);
WRITELN('=====:29);
FOR INDEX:=0 TO DAUER DO
  WRITELN(INDEX;22,HISTO(INDEX);7);
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
WRITELN('LEGENDE:');
WRITELN('ANZAHL ANZAHL DER WERKZEUGE');
WRITELN('LD LEBENSDAUER IN STUNDEN,GERUNDET')
END. (* WERKZEUG *)

```

```

21 27
22 11
23 0
24 1
25 0

```

```

=====
LEGENDE:
ANZAHL ANZAHL DER WERKZEUGE
LD LEBENSDAUER IN STUNDEN,GERUNDET

```

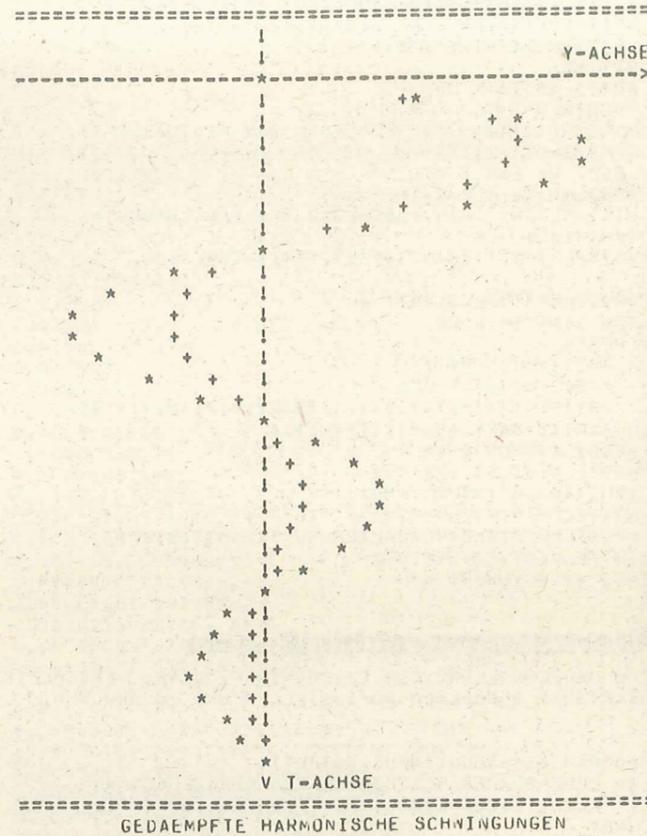
Grafik gedämpfter Schwingungen

$Y=A \cdot \exp(\alpha \cdot t) \cdot \sin(\omega \cdot t + \phi)$, wobei $\alpha < 0$, stellt eine gedämpfte harmonische Schwingung dar. Die Funktionsverläufe sind fuer zwei Werte von α graphisch darzustellen.

```

PROGRAM SCHWINGE(OUTPUT);
(* HARMONISCHE SCHWINGUNGEN M,VERSCH,DAEMPfung *)
(* FELD,FOR=ANWEISUNG,STANDARDfUNKTIONEN *)
CONST
AM=1; (* AMPLITUDEN=fAKTOR. *)
ALFA=-1;
ALF1=-2;
FI=0; (* ANfangSPHASE. *)
OMEGA=6.28318; (* SCHWINGUNGS-fREQUENZ. *)
D=0.0625; (* 16 STUETZWERTE IN T...T+1. *)
S=32; (* DISKRETE ORDINATEN=WERTE. *)
H=20; (* ABSTAND DER T-ACHSE VOM LINKEN RAND. *)
ANZ=32; (* ANZAHL DER STUETZSTELLEN. *)
S2=64;
VAR
I,J,M,N,ORD:INTEGER;
Y,T:REAL;
A:ARRAY[1..S2] OF CHAR;
BEGIN
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
FOR I:=1 TO S2 DO A[I]:=' ';
WRITELN('!':H);
WRITELN('!':H,'Y-ACHSE':50-H);
FOR I:=1 TO 49 DO IF I=H THEN WRITE('*')
ELSE WRITE(' ');
WRITELN('>');
FOR I:=1 TO ANZ DO
BEGIN
T:=D*I;
Y:=AM*EXP(ALF1*T)*SIN(OMEGA*T+FI);
A[H]:='!';
N:=ROUND(S*Y)+H;
A[N]:='+';
Y:=AM*EXP(ALFA*T)*SIN(OMEGA*T+FI);
M:=ROUND(S*Y)+H;
A[M]:='*';
IF M<H THEN ORD:=H ELSE ORD:=M;
FOR J:=1 TO ORD DO WRITE(A[J]); WRITELN;
A[N]:='!'; A[M]:='!';
END; (* FOR I *)
WRITELN('V':H,'T-ACHSE');
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
WRITELN('GEDAEMPfTE HARMONISCHE SCHWINGUNGEN')
END. (* SCHWINGE *)

```



Arbeitstage

IM RAHMEN EINER INNERBETRIEBLICHEN ARBEITSZEITREGLUNG IST EINE UEBERSICHT UEBER DIE TAGE DER WOCHE ZU DRUCKEN.

```

PROGRAM TAGE(OUTPUT);
(* TAGE DER WOCHE *)
(* CASE=ANWEISUNG,MENGENTYP,MENGENOPERATOREN *)
TYPE
TAGE=(MO,DI,MI,DT,FR,SA,SO);
WOCHE=SET OF TAGE;
VAR
T:TAGE;
ARBEITSTAGE,FREIETAGE,WOCHENTAGE:WOCHE;
BEGIN
WRITELN('UEBERSICHT UEBER DIE TAGE DER WOCHE: ');
WRITELN('==','==':48);

```

```

UEBERSICHT UEBER DIE TAGE DER WOCHE:
--
FUER ALLE BEREICHE, DIE NICHT IM SCHICHTBETRIEB
ARBEITEN GILT FOLGENDE REGLUNG:
--
MONTAG IST ARBEITSTAG
DIENSTAG IST ARBEITSTAG
MITTWOCH IST ARBEITSTAG
DONNERSTAG IST ARBEITSTAG
FREITAG IST ARBEITSTAG
SONNABEND IST EIN FREIER TAG
SONNTAG IST EIN FREIER TAG

```

```

WRITELN('FUER ALLE BEREICHE, DIE NICHT IM SCHICHTBETRIEB');
WRITELN('ARBEITEN GILT FOLGENDE REGLUNG:');
WRITELN('==','==':48);
ARBEITSTAGE:={MO,..FR};
FREIETAGE:={SA,..SO};
WOCHENTAGE:=ARBEITSTAGE + FREIETAGE;
FOR T:=MO TO SO DO
BEGIN
CASE T OF
MO:WRITE('MONTAG':17); (* T IST SELEKTOR *)
DI:WRITE('DIENSTAG':19);
MI:WRITE('MITTWOCH':19);
DT:WRITE('DONNERSTAG':21);
FR:WRITE('FREITAG':18);
SA:WRITE('SONNABEND':20);
SO:WRITE('SONNTAG':18);
END; (* OF CASE *)
IF T IN ARBEITSTAGE THEN
WRITELN(' IST ARBEITSTAG');
IF T IN FREIETAGE THEN
WRITELN(' IST EIN FREIER TAG');
END; (* FOR T *)
WRITELN('==','==':48);
END. (* TAGE *)

```

Zwei Schaltungen

ZWEI SCHALTUNGEN WERDEN DURCH DIE LOGISCHEN FUNKTIONEN Y BZW. Z BESCHRIEBEN. ES IST ZU ZEIGEN, DASS DIE BEIDEN FUNKTIONEN Y UND Z AEQUIVALENT SIND UND DESHALB HEIDE SCHALTUNGEN GLEICHE VERHALTENSWEISEN BESITZEN. WERDEN DIE UNABHAENIGEN VARIABLEN DURCH P,Q,R UND DIE OPERATOREN AND DURCH +, OR DURCH V UND NEGATION DURCH - BEZEICHNET, SO SIND $Y=(P \uparrow Q \uparrow R) \vee (P \uparrow Q \uparrow R)$ UND $Z=(P \uparrow Q) \vee R$.

```

PROGRAM AEQUIVAL(OUTPUT);
(* AEQUIVALENZ VON LOG. FUNKTIONEN *)
(* FOR=ANWEISUNG,BOOLESCHE OPERATOREN *)
VAR
P,Q,R,Y,Y1,Y2,Y3,Y4,Y5,Z,T,TEST:BOOLEAN;
BEGIN
WRITELN('ZWEI AEQUIVALENTE LOGISCHE FUNKTIONEN');
WRITELN('ES IST Y=Y1 OR Y2 OR Y3 OR Y4 OR Y5,');
WRITELN('MIT');
WRITELN('Y1=P AND Q AND R');
WRITELN('Y2=P AND NOT Q AND R');
WRITELN('Y3=P AND NOT Q AND NOT R');
WRITELN('Y4=NOT P AND Q AND R');
WRITELN('Y5=NOT P AND NOT Q AND R UND');
WRITELN('Z=(P AND NOT Q) OR R.');
WRITELN('DIE WERTETABELLE HAT FOLGENDE GESTALT');
WRITELN('P Q R Y Z Y=Z');
FOR P:=FALSE TO TRUE DO
FOR Q:=FALSE TO TRUE DO
FOR R:=FALSE TO TRUE DO
BEGIN
Y1:=P AND Q AND R;
Y2:=P AND NOT Q AND R;
Y3:=P AND NOT Q AND NOT R;
Y4:=NOT P AND Q AND R;
Y5:=NOT P AND NOT Q AND R;
Y:=Y1 OR Y2 OR Y3 OR Y4 OR Y5;
Z:=(P AND NOT Q) OR R;
TEST:=Y=Z;
WRITELN(P;5,Q;6,R;6,Y;7,Z;7,TEST;6);
END;
WRITELN('BEIDE FUNKTIONEN');
IF T THEN WRITELN('SIND AEQUIVALENT !');
ELSE WRITELN('NICHT AEQUIVALENT !');
WRITELN('==','==':48);
END. (* AEQUIVAL *)

```

ZWEI AEQUIVALENTE LOGISCHE FUNKTIONEN

ES IST Y=Y1 OR Y2 OR Y3 OR Y4 OR Y5, MIT

Y1=P AND Q AND R
Y2=P AND NOT Q AND R
Y3=P AND NOT Q AND NOT R
Y4=NOT P AND Q AND R
Y5=NOT P AND NOT Q AND R UND
Z=(P AND NOT Q) OR R.

DIE WERTETABELLE HAT FOLGENDE GESTALT

P	Q	R	Y	Z	Y=Z
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	FALSE	FALSE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

BEIDE FUNKTIONEN SIND AEQUIVALENT !

Operationen mit eindimensionalen Feldern

VERSCHIEDENE TYPEN EINDIMENSIONALER FELDER SIND ZU DEFINIEREN UND MIT ERLAUBTEN OPERATOREN ZU NUTZEN !

```
PROGRAM FELDER(INPUT,OUTPUT);
(* OPERATIONEN MIT EINDIM. FELDERN *)
(* FELDTYPEN, FOR=ANWEISUNG *)
CONST
L=10;
TYPE
VEKTOR1=ARRAY[1..L] OF REAL;
VEKTOR2=ARRAY[1..10] OF ALFA;
VEKTOR3=ARRAY['A'..'Z'] OF CHAR;
VAR
BZ:CHAR; I,J,K:INTEGER; SPR:REAL;
V1,V2:VEKTOR1; A1,A2:VEKTOR2; C1,C2:VEKTOR3;
BEGIN
FOR I:=1 TO 50 DO WRITE('$'); WRITELN;
WRITELN('EINDIMENSIONALE FELDER' : 36);
FOR I:=1 TO 50 DO WRITE('$'); WRITELN;
WRITELN('VEKTOREN" MIT REELLEN KOMONENTEN : ');
WRITELN('WERTE DER FELDELEMENTE VON V1 : 50);
FOR I:=1 TO L DO
BEGIN
V1[I]:=I; (* INDEXWERT WIRD V1[I] ZUGEWIESEN *)
WRITE(V1[I]:5:1);
END; WRITELN; WRITELN(' '); V2:=V1; SPR:=0;
WRITELN('WERTZUWEISUNG AN V2 DURCH V2:=V1 ');
FOR I:=1 TO L DO WRITE(V2[I]:5:1); WRITELN;
WRITELN('SKALARES PRODUKT DER VEKTOREN V1, V2:');
FOR I:=1 TO L DO SPR:=SPR+V1[I]*V2[I];
WRITELN('SPR:46:2);
WRITELN('FELDER MIT ELEMENTEN VOM ALFA-TYP : ');
WRITELN('FELDELEMENTE VON A1 SIND:50);
FOR I:=1 TO 10 DO
BEGIN
FOR J:=1 TO 8 DO READ(A1[I,J]); READLN;
WRITE(A1[I]:5);
END; (* FOR I *) WRITELN; A2:=A1;
WRITELN('WERTZUWEISUNG AN A2 DURCH A2:=A1 ');
FOR I:=1 TO 10 DO WRITE(A2[I]:5); WRITELN;
WRITELN('FELDER MIT ELEMENTEN VOM CHAR-TYP : ');
WRITELN('WERTE DER ELEMENTE VON C1 : 50);
FOR BZ:='A' TO 'Z' DO C1[BZ]:=BZ;
FOR BZ:='A' TO 'Z' DO WRITE(C1[BZ]); WRITELN;
WRITELN('FELD C2:= FELD C1. '); C2:=C1;
WRITELN('VERGLEICH ZWEIER FELDER VOM CHAR-TYP ');
IF C1=C2 THEN WRITELN('FELD C1 = FELD C2');
ELSE WRITELN('FELD C1 <> FELD C2');
WRITELN('BEI V1,V2 UND A1,A2 FUEHRT TEST AUF GL',
'EICHHEIT ZUM');
WRITELN('FEHLER: "ILLEGALER OPERANDENTYP" !');
WRITELN('AUSGEWAHLTE FELDELEMENTE VON C2 SIND');
FOR BZ:='A' TO 'I' DO WRITE(C2[BZ]);
FOR BZ:='J' TO 'R' DO WRITE(C2[BZ]);
FOR BZ:='S' TO 'Z' DO WRITE(C2[BZ]); WRITELN;
WRITELN('G E S P E R R T E R DRUCK:50);
FOR K:=1 TO 5 DO
BEGIN
FOR BZ:='A' TO 'I' DO WRITE(C1[BZ]:K); WRITELN
END; (* FOR K *)
FOR I:=1 TO 50 DO WRITE('$'); WRITELN;
END. (* FELDER *)
```

Vierpolmatrizen

EINE ELEKTRISCHE SCHALTUNG ZUR UEBERTRAGUNG ELEKTRISCHER LEISTUNG HEISZT "VIERPOL". DIE ALLGEMEINE FORM DER VIERPOLGLEICHUNGEN MIT 4 VERSCHIEDENEN VONEINANDER UNABHAENIGEN VIERPOLKONSTANTEN KANN ALS MATRIZENGLEICHUNG DARGESTELLT WERDEN: $U = w * J$, WOBEI U DIE SPANNUNGS-, W DIE WIDERSTANDS- UND J DIE STROM-MATRIX BEZEICHNEN. FUER EINE GEGEBENE WIDERSTANDSMATRIX W UND VERSCHIEDENE J IST U ZU BERECHNEN!

```
PROGRAM VIERPOL(INPUT,OUTPUT);
(* VIERPOLMATRIZEN *)
(* FELDTYPEN, FOR=ANWEISUNG *)
CONST
W11='EINGANGS=LEERLAUFWIDERSTAND';
W21='KERNWIDERSTAND VORWAERTS';
W12='KERNWIDERSTAND RUECKWAERTS';
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
EINDIMENSIONALE FELDER
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
"VEKTOREN" MIT REELLEN KOMONENTEN :
WERTE DER FELDELEMENTE VON V1:
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
.
WERTZUWEISUNG AN V2 DURCH V2:=V1
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
SKALARES PRODUKT DER VEKTOREN V1, V2:
385.00
FELDER MIT ELEMENTEN VOM ALFA-TYP :
FELDELEMENTE VON A1 SIND:
PITT HANS RUDI EDDA RUTH ANNE ROSI UWE JENS TONI
WERTZUWEISUNG AN A2 DURCH A2:=A1
PITT HANS RUDI EDDA RUTH ANNE ROSI UWE JENS TONI
FELDER MIT ELEMENTEN VOM CHAR-TYP :
WERTE DER ELEMENTE VON C1 :
ABCDEFGHI JKLMNOPQR \ STUVWXYZ
FELD C2:= FELD C1.
VERGLEICH ZWEIER FELDER VOM CHAR-TYP
FELD C1 = FELD C2
BEI V1,V2 UND A1,A2 FUEHRT TEST AUF GLEICHHEIT ZUM
FEHLER: "ILLEGALER OPERANDENTYP" !
AUSGEWAHLTE FELDELEMENTE VON C2 SIND
ABCDEFGHIJKLMNPOQRSTUVWXYZ
G E S P E R R T E R DRUCK:
ABCDEFGHI
A B C D E F G H I
A B C D E F G H I
A B C D E F G H I
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
*****
VIERPOLMATRIZEN ZUR BESCHREIBUNG
ELEKTRISCHER SCHALTUNGEN.
*****
DIE WERTE DER ELEMENTE DER WIDERSTANDSMATRIX SIND:
EINGANGS=LEERLAUFWIDERSTAND W[1,1]= 850.55
```

```
W22='AUSGANGS=LEERLAUFWIDERSTAND';
VAR
I,L,M:INTEGER;
U,W,J:ARRAY[1..2,1..2] OF REAL;
BEGIN
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
WRITELN('V I E R P O L M A T R I Z E N ZUR BES',
'CHREIBUNG ');
WRITELN('ELEKTRISCHER SCHALTUNGEN.1:50);
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
FOR I:=1 TO 2 DO READLN(W[I,1],W[I,2]);
WRITELN('DIE WERTE DER ELEMENTE DER WIDERSTANDS',
'MATRIX SIND:'); WRITELN(' ');
WRITELN('W[1,1]=',W[1,1]:7:2);
WRITELN('W[1,2]=',W[1,2]:7:2);
WRITELN('W[2,1]=',W[2,1]:7:2);
WRITELN('W[2,2]=',W[2,2]:7:2);
READLN(M);
FOR I:=1 TO M DO
BEGIN
WRITELN('I:1,');
WRITELN('DIE WERTE DER ELEMENTE DER STROMMATR',
'IX J SIND:');
READLN(J[1,1],J[2,1]);
J[1,2]:=0; J[2,2]:=0;
WRITELN('J[1,1]=',J[1,1]:7:2,' J[1,2]= 0');
WRITELN('J[2,1]=',J[2,1]:7:2,' J[2,2]= 0');
WRITELN('** FOLGENDE WERTE DER ELEMENTE DER S',
'PANNUNGSMATRIX');
WRITELN('WURDEN BERECHNET:');
U[1,1]:=W[1,1]*J[1,1] + W[1,2]*J[2,1];
U[2,1]:=W[2,1]*J[1,1] + W[2,2]*J[2,1];
U[1,2]:=0; U[2,2]:=0;
WRITELN('U[1,1]=',U[1,1]:7:2,' U[1,2]= 0');
WRITELN('U[2,1]=',U[2,1]:7:2,' U[2,2]= 0');
END; (* FOR L *)
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
END. (* VIERPOL *)
```

Inverse elementare Funktionen

ES SIND ZWEI ZU EINANDER INVERSE ELEMENTARE FUNKTIONEN ZU DEFINIEREN, WELCHE KONSTANTEN, DIE WAHRHEITSWERTE DARSTELLEN, ZAHLNWERTE ZUORDNEN UND UMGEKEHRT.

```
PROGRAM FLEFUNKT(OUTPUT);
(* ZUORDNUNGEN VON KONSTANTEN *)
(* FUNKTION,CASE=ANWEISUNG, FOR=ANWEISUNG *)
TYPE
ZIFFERN=0..1;
VAR
DUAL:ZIFFERN;
LOGI,BOOL:BOOLEAN;
FUNCTION INTTOLOG(ZAHL:ZIFFERN):BOOLEAN;
BEGIN
CASE ZAHL OF
0:INTTOLOG:=FALSE;
1:INTTOLOG:=TRUE
END
END; (* INTTOLOG *)
FUNCTION LOGTOINT(LOG:BOOLEAN):ZIFFERN;
BEGIN
CASE LOG OF
TRUE:LOGTOINT:=1;
FALSE:LOGTOINT:=0
END
END; (* LOGTOINT *) (* ROUTINENDEKLARATIONSTEIL *)
BEGIN
WRITELN('EINFACHE FUNKTIONEN, DIE EINEN ZUSAMME',
'NHANG "DUALE");
WRITELN('LOGISCHE" KONSTANTEN HERSTELLEN');
FOR DUAL:=0 TO 1 DO
WRITE('*****'); WRITELN;
FOR DUAL:=0 TO 1 DO
WRITELN('DUAL:1, WIRD',INTTOLOG(DUAL):6,' ZUGE',
'ORDNET. '); WRITELN('*****:31);
FOR LOGI:=FALSE TO TRUE DO
WRITELN('LOGI:5, WIRD',LOGTOINT(LOGI):2,' ZUGE',
'ORDNET. '); WRITELN('*****:31);
FOR BOOL:=FALSE TO TRUE DO
BEGIN
LOGI:=INTTOLOG(LOGTOINT(BOOL));
```

```
KERNWIDERSTAND RUECKWAERTS W[1,2]= 999.95
KERNWIDERSTAND VORWAERTS W[2,1]= 885.75
AUSGANGS=LEERLAUFWIDERSTAND W[2,2]= 900.50
1.
DIE WERTE DER ELEMENTE DER STROMMATRIX J SIND:
J[1,1]= 10.00 J[1,2]= 0
J[2,1]= 10.00 J[2,2]= 0
** FOLGENDE WERTE DER ELEMENTE DER SPANNUNGSMATRIX
WURDEN BERECHNET:
U[1,1]= 18504.97 U[1,2]= 0
U[2,1]= 17862.48 U[2,2]= 0
2.
DIE WERTE DER ELEMENTE DER STROMMATRIX J SIND:
J[1,1]= 25.00 J[1,2]= 0
J[2,1]= 20.50 J[2,2]= 0
** FOLGENDE WERTE DER ELEMENTE DER SPANNUNGSMATRIX
WURDEN BERECHNET:
U[1,1]= 41762.67 U[1,2]= 0
U[2,1]= 40603.94 U[2,2]= 0
3.
DIE WERTE DER ELEMENTE DER STROMMATRIX J SIND:
J[1,1]= 10.25 J[1,2]= 0
J[2,1]= 12.75 J[2,2]= 0
** FOLGENDE WERTE DER ELEMENTE DER SPANNUNGSMATRIX
WURDEN BERECHNET:
U[1,1]= 21467.48 U[1,2]= 0
U[2,1]= 20560.28 U[2,2]= 0
*****
```

```

WRITELN('FUER BOOL=' , BOOL:6, ' IST INTTOLOG(LO',
'GTOINT(BOOL))=' ); WRITELN(LOGI:50)
END;
FOR DUAL:=0 TO 1 DO
WRITELN('FUER DUAL=' , DUAL:2, ' IST LUGTOINT(INT',
'TOLOG(DUAL))=' , LUGTOINT(INTTOLOG(DUAL))
:2);
FOR DUAL:=0 TO 1 DO
WRITE('#####'); WRITELN;
END. (* ELEFUNKT *)

```

Darstellung einer Matrix

ZUR BESCHREIBUNG LINEARER ABHAENGIKHEITEN ZWISCHEN MEHREREN GROESZEN WERDEN MATRIZEN VERWENDET. DIESE KOENNEN DURCH ZWEIDIMENSIONALE FELDER ABGEBILDET WERDEN. DIE WERTE DER ELEMENTE EINER MATRIX MIT M ZEILEN UND N SPALTEN SIND EINZULESEN UND IN UEBERSICHTLICHER FORM ZU DRUCKEN!

```

PROGRAM MATRIX(INPUT,OUTPUT);
(* ZWEIDIMENSIONALES FELDER AUSGEBEN *)
(* FELDER, FOR=ANWEISUNG, VERBUNDANWEISUNG, WRITELN *)
CONST
M=13;
N=10;
VAR
I,J:INTEGER;
A:ARRAY[1..M,1..N] OF REAL;
BEGIN
FOR I:=1 TO 50 DO WRITE('*'); WRITELN;
WRITELN('MATRIX MIT M ZEILEN UND N SPALTEN');
FOR I:=1 TO 50 DO WRITE('*'); WRITELN;
FOR I:=1 TO M DO
BEGIN
FOR J:=1 TO N DO READ(A[I,J]);
READLN
END; (* FOR I *)
WRITELN(' ',50);
WRITE('SPALTE ',10);
FOR I:=1 TO N DO WRITE(I:4); WRITELN;
WRITE('ZEILE ');
FOR I:=1 TO 40 DO WRITE(' '); WRITELN;
FOR I:=1 TO M DO
BEGIN
WRITE(I:5, ' ',5);
FOR J:=1 TO N DO WRITE(A[I,J]:4:1); WRITELN;
END; (* FOR I *)
WRITELN(' ',50);
FOR I:=1 TO 50 DO WRITE('*'); WRITELN;
END. (* MATRIX *)

```

```

*****
MATRIX MIT M ZEILEN UND N SPALTEN
*****

```

SPALTE	1	2	3	4	5	6	7	8	9	10
1	1.0	1.5	3.7	4.1	9.0	6.0	8.7	7.5	9.9	2.2
2	4.6	4.7	0	8.0	7.4	8.3	9.1	7.0	8.0	4.1
3	6.1	7.4	0.1	1.4	0.2	0.9	1.0	2.0	0.8	1.5
4	2.2	9.9	9.5	5.7	7.8	0.6	0.9	1.4	7.3	5.1
5	1.5	6.4	7.4	0	0.8	4.7	3.8	1.9	0.7	0.8
6	0.9	9.5	0.7	8.7	9.5	9.5	8.3	9.1	0.4	8.1
7	9.9	9.5	5.7	8.7	9.5	9.9	8.3	9.1	0.4	8.8
8	1.6	4.7	1.0	4.1	2.0	9.0	5.1	8.0	0.2	1.0
9	6.9	1.8	1.7	8.1	8.3	9.8	9.7	1.5	4.3	1.1
10	1.9	1.8	1.7	8.1	8.3	9.8	9.7	1.5	0.3	9.1
11	7.6	4.7	1.0	4.1	2.0	9.0	5.1	8.0	0.2	8.0
12	4.7	1.1	0.1	5.7	0.8	4.7	9.1	8.4	2.4	7.7
13	2.2	0.8	2.7	0.9	4.7	2.0	8.7	0.1	0.8	2.5

Tabellierung von Größen des Kreises

DIE GROESZEN DES KREISES UMFANG UND FLAECHEINHALT SIND IN ABHAENGIKHEIT VON DEN WERTEN DES RADIUS R IN EINER UEBERSICHTLICHEN FORM ZU TABELLIEREN.

```

PROGRAM KREISTAB(OUTPUT);
(* TABELLE VON GROESZEN DES KREISES *)
(* FUNKTIONSDOKUMENTATION, FUNKTIONSAUFRUF *)
CONST
PI=3.141592654;
ANZ=24;
ST='-----';
VAR
I:INTEGER;
FUNCTION UMFANG(RADIUS:REAL):REAL;
BEGIN
UMFANG:=2*PI*RADIUS
END; (* UMFANG *)
FUNCTION FLAECHE(RADIUS:REAL):REAL;
BEGIN
FLAECHE:=PI*SQR(RADIUS)
END; (* FLAECHE *)
(* ENDE DES ROUTINENDEKLARATIONSTEILS *)
BEGIN
WRITELN(ST,ST,ST:10);
WRITELN('UMFAENGE UND FLAECHEINHALTE VON KR:40,
'EISEN');
WRITELN(ST,ST,ST:10);
WRITELN('RADIUS UMFANG FLAECHE:38);

```

```

-----
UMFAENGE UND FLAECHEINHALTE VON KREISEN
-----

```

RADIUS	UMFANG	FLAECHE
1	6.2832	3.1416
2	12.5663	12.5663
3	18.8495	28.2743
4	25.1327	50.2654
5	31.4159	78.5397
6	37.6991	113.0971
7	43.9823	153.9378
8	50.2654	201.0617
9	56.5486	254.4686
10	62.8318	314.1587
11	69.1150	380.1320
12	75.3982	452.3886
13	81.6813	530.9283
14	87.9645	615.7513
15	94.2477	706.8575
16	100.5308	804.2468
17	106.8139	907.9191
18	113.0971	1017.8744
19	119.3803	1134.1130

```

WRITELN(ST:32,ST:6);
FOR I:=1 TO ANZ DO WRITELN(I:18,UMFANG(I):10:4,
(* AUFRUFE DER FUNKTIONEN ! *) FLAECHE(I):10:4);
WRITELN(ST,ST,ST:10);
END. (* KREISTAB *)

```

20	125.6635	1256.6345
21	131.9467	1385.4408
22	138.2299	1520.5287
23	144.5131	1661.9002
24	150.7962	1809.5546

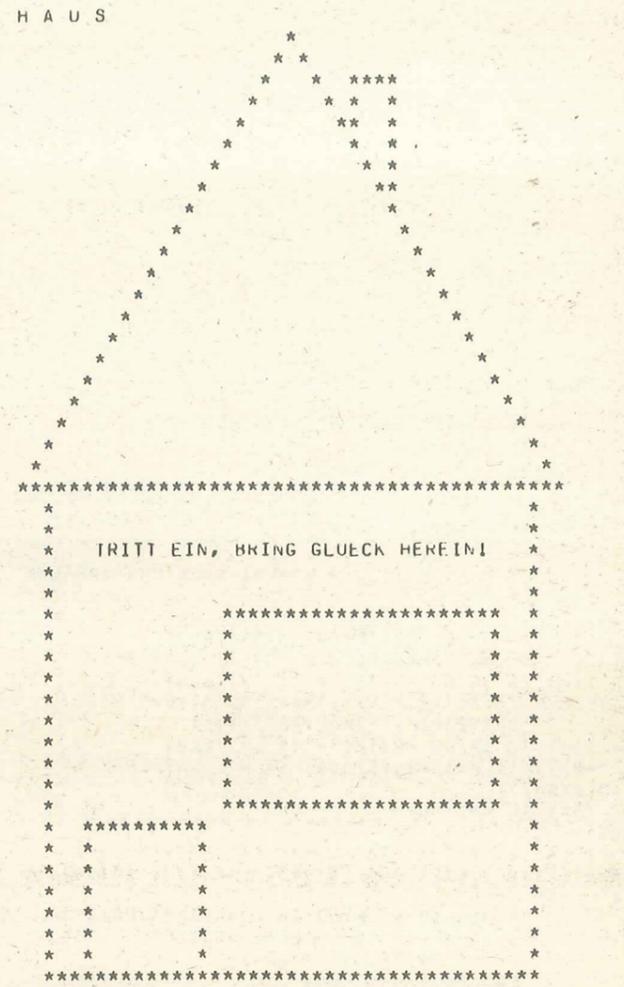
Haus

DIE SKIZZE EINES HAUSES IST ZU DRUCKEN. DAS HAUS SOLL SO DARGESTELLT WERDEN, DASS EIN FENSTER, EINE TUER UND EIN SCHORNSTEIN ZU SEHEN SIND. AUSZERDEM IST UEBER DEM FENSTER EIN SPRUCH VORZUSEHEN.

```

PROGRAM HAUS(OUTPUT);
(* SKIZZE EINES HAUSES *)
(* FOR=ANWEISUNG, IF=ANWEISUNG, *)
VAR
I,N:INTEGER;
BEGIN
WRITELN('H A U S');
N:=22; (* ABSTAND VOM LINKEN DRUCKLISTENRAND *)
WRITELN('*:N); (* DACHFIRST *)
FOR I:=1 TO 20 DO
BEGIN
WRITE('*:N-I, '*:2*I);
IF I=2 THEN WRITE(' - ****');
IF (I=3) OR (I=4) THEN WRITE('*:5-I, '*:3);
IF (I>4) AND (I<8) THEN WRITE('*:8-I);
WRITELN
END;
WRITE('*:N-21);
FOR I:=1 TO 42 DO WRITE('*');
WRITELN; (* DACH MIT SCHORNSTEIN FERTIG *)
WRITELN('*:N-19, '*:38);
WRITELN('*:N-19, '*:38);
WRITE('*:N-19, ' TRITT EIN, BRING GLUECK HE');
WRITELN('REIN! *');
WRITELN('*:N-19, '*:38);
WRITELN('*:N-19, '*:38);
WRITE('*:N-19);
WRITE('*:14);
FOR I:=1 TO 21 DO WRITE('*');
WRITELN('*:13);
FOR I:=1 TO 8 DO
WRITELN('*:N-19, '*:14, '*:21, '*:3);
WRITE('*:N-19);
WRITE('*:14);
FOR I:=1 TO 21 DO WRITE('*');
WRITELN('*:13);
WRITE('*:N-19, '*:3);
FOR I:=1 TO 9 DO WRITE('*');
WRITELN('*:26);
FOR I:=1 TO 6 DO
WRITELN('*:N-19, '*:3, '*:9, '*:26);
WRITE('*:N-19);
FOR I:=1 TO 38 DO WRITE('*');
WRITELN; (* GRUNDLINIE DES HAUSES *)
END. (* HAUS *)

```



Anwendung einer Marke

DIE ANWENDUNG EINER MARKE IN VERBINDUNG MIT EINER GOTO=ANWEISUNG ISI DURCH EIN EINFACHES PROGRAMM EINZUFUEHREN!

```

PROGRAM MARKE(OUTPUT);
(* UNBEDINGTER SPRUNG *)
(* MARKE, GOTO=ANWEISUNG *)
LABEL 1;
BEGIN
##### GOTO=ANWEISUNG UND MARKE
### NUR GOTO 1 FUEHRT ZU DIESEM DRUCK !

```

```

WRITE('##### GOTO=ANWEISUNG UND MARKE #####');
GOTO 1;
WRITELN('KEIN DRUCK!');
1:WRITELN(' NUR GOTO 1 FUEHRT ZU DIESEM DRUCK !');
END. (* MARKE *)

```

Rationalisierung der Programmierung

ZUR UNTERSTÜTZUNG DER EIGENEN ARBEIT IST EINE "SAMMLUNG" SELBST DEKLARierter EINFACHER FUNKTIONEN AUFZUBAUEN UND DIE ANWENDUNG FÜR BELIEBIGE EINGABEDATEN VORZUFÜHREN!

```
PROGRAM EIGENBAU(INPUT,OUTPUT);
(* FUNKTIONEN VON VERSCHIEDENEM TYP *)
(* FUNKTIONSDEKLARATION *)
VAR
  I,J,ANZ,MI:INTEGER; Y,MR:REAL;
FUNCTION SIGNUM(X:REAL):INTEGER;
BEGIN
  IF X=0 THEN SIGNUM:=0
  ELSE IF X>0 THEN SIGNUM:=1
  ELSE SIGNUM:=-1
END; (* SIGNUM *)
FUNCTION MAX(X,Y:REAL):REAL;
BEGIN
  IF X>Y THEN MAX:=X ELSE MAX:=Y
END; (* MAX *)
FUNCTION MAXI(X,Y:INTEGER):INTEGER;
BEGIN
  IF X>Y THEN MAXI:=X ELSE MAXI:=Y
END; (* MAXI *)
FUNCTION FAK(N:INTEGER):INTEGER;
VAR
  I,F:INTEGER;
BEGIN
  F:=1;
  FOR I:=1 TO N DO F:=F*I; FAK:=F
END; (* FAK *)
FUNCTION PRIM(I:INTEGER):BOOLEAN;
VAR
  L:INTEGER;
BEGIN
  PRIM:=TRUE;
  IF I<2 THEN PRIM:=FALSE;
  IF I>2 THEN FOR L:=2 TO I-1 DO
    IF (I DIV L)*L=I THEN PRIM:=FALSE
  END; (* PRIM *)
BEGIN
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  WRITELN('SELBST DEKLARIERTE FUNKTIONEN':39);
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  WRITELN(' I ', 'Y':8, ' VZ':8, ' MAX':8, ' J':5, ' MAXI':
    ' PRIMZAHL', 'FAKULTAET':11);
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  MI:=MAXINT; MR:=-1.0E+30; (* ANFANGSWERTE *)
  READLN(ANZ);
  FOR I:=1 TO ANZ DO
    BEGIN
      READLN(Y,J); MR:=MAX(MR,Y); MI:=MAXI(MI,J);
      WRITE(I:2,Y:8:2,SIGNUM(Y):3,MR:8:2,J:5,MI:4);
      IF J>0 THEN
        BEGIN IF PRIM(J) THEN WRITE(' PRIM')
          ELSE WRITE(' ');
          WRITELN(FAK(J):15)
        END ELSE WRITELN;
    END; (* FÜR I *)
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* EIGENBAU *)
```

```
=====
SELBST DEKLARIERTE FUNKTIONEN
=====
I      Y VZ      MAX      J MAX PRIMZAHL FAKULTAET
-----
1 -125.75 -1 -125.75  5  5 PRIM          120
2 1010.10  1 1010.10  2  5 PRIM           2
3 1234.56  1 1234.56  3  5 PRIM           6
4 -10.50 -1 1234.56  1  5                1
5 4321.00  1 4321.00 -1  5                1
6 120.00  1 4321.00  4  5                24
7  2.00  1 4321.00  8  8                40320
8 100.00  1 4321.00 -30 8                362880
9  1.00  1 4321.00  9  9                362880
10  3.00  1 4321.00 -7  9                5040
11  4.00  1 4321.00  7  9 PRIM          5040
12  0  0 4321.00 10 10          362880
13 133.00  1 4321.00 -200 10          1
14 2050.00  1 4321.00  0 10          720
15 666.00  1 4321.00  6 10          1
16 7070.00  1 7070.00 -2 10          1
17 333.33  1 7070.00 -5 10          1
18 222.50  1 7070.00 -6 10          1
19 9009.55  1 9009.55 -8 10          1
20 -575.96 -1 9009.55 11 11 PRIM      39916800
=====
```

Dreiecksflächen

ZAHLENTRIPEL SIND ZU PRUEFEN, OB SIE MASZZAHLEN DER SEITENLAENGEN VON DREIECKEN SIND. WENN JA, DANN IST DER FLAECHEINHALT ZU ERMITTELN.

```
PROGRAM DREIECKE(INPUT,OUTPUT);
(* DREIECKSFLAECHE *)
(* FUNKTION,FUNKTIONSAUFRUF,LOKAL *)
VAR
  I:INTEGER;
  A,B,C,F:REAL;
  LOG:BOOLEAN;
FUNCTION SEITEST(A,B,C:REAL):BOOLEAN;
BEGIN
  SEITEST:=(A<=B+C) AND (B<=A+C) AND (C<=A+B)
END; (* SEITEST *)
FUNCTION FLAECHE3(A,B,C:REAL):REAL;
```

```
=====
DREIECKSFLAECHE
=====
1.
GEGEBEN SIND DIE LAENGEN DER SEITEN
A = 30.00 M
B = 40.00 M
C = 50.00 M
DADURCH IST EIN DREIECK FESTGELEGT.
BERECHNET WURDE DIE FLAECHE = 600.00 M*M
2.
GEGEBEN SIND DIE LAENGEN DER SEITEN
```

```
VAR
  U:REAL;
BEGIN
  U:=(A+B+C)/2;
  FLAECHE3:=SQRT(U*(U-A)*(U-B)*(U-C))
END; (* FLAECHE3 *)
BEGIN
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  WRITELN('DREIECKSFLAECHE':33);
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  FOR I:=1 TO 4 DO
    BEGIN
      READLN(A,B,C); WRITELN(I:25,' ');
      WRITELN('GEGEBEN SIND DIE LAENGEN DER SEITEN');
      WRITELN('A =',A:8:2,' M');
      WRITELN('B =',B:8:2,' M');
      WRITELN('C =',C:8:2,' M');
      WRITE('DADURCH IST ');
      IF SEITEST(A,B,C) THEN
        BEGIN
          WRITELN('EIN DREIECK FESTGELEGT. ');
          WRITELN('BERECHNET WURDE DIE FLAECHE =',
            FLAECHE3(A,B,C):8:2,' M*M');
        END ELSE WRITELN('KEIN DREIECK FESTGELEGT. ');
    END; (* FOR I *)
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* DREIECK *)
```

```
A = 30.00 M
B = 50.00 M
C = 90.00 M
DADURCH IST KEIN DREIECK FESTGELEGT!
3.
GEGEBEN SIND DIE LAENGEN DER SEITEN
A = 100.00 M
B = 200.00 M
C = 300.00 M
DADURCH IST EIN DREIECK FESTGELEGT.
BERECHNET WURDE DIE FLAECHE = 0 M*M
4.
GEGEBEN SIND DIE LAENGEN DER SEITEN
A = 45.00 M
B = 55.00 M
C = 65.00 M
DADURCH IST EIN DREIECK FESTGELEGT.
BERECHNET WURDE DIE FLAECHE = 1220.19 M*M
=====
```

Vergleich zweier Algorithmen

DIE ALGORITHMEN ZUR ITERATIVEN UND REKURSIVEN BERECHNUNG DER FAKULTAET SIND ZU VERGLEICHEN. DIE UEBEREINSTIMMUNG DER ERGEBNISSE IST ZU ZEIGEN.

```
PROGRAM FAKULTAE(OUTPUT);
(* REKURSIVE U.ITERATIVE FAKULTAETSBERECHNUNG *)
(* FUNKTION,LOKAL *)
VAR
  I,FR,FI:INTEGER;
FUNCTION FAK(K:INTEGER):INTEGER;
VAR
  F,I:INTEGER;
BEGIN
  F:=1;
  FOR I:=2 TO K DO F:=F*I;
  FAK:=F
END; (* FAK *)
FUNCTION FAKR(K:INTEGER):INTEGER;
BEGIN
  IF K>1 THEN FAKR:=K*FAKR(K-1)
  ELSE FAKR:=1
END; (* FAKR *)
BEGIN
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  WRITELN('REKURSIVE UND ITERATIVE FAKULTAETS-
    BERECHNUNG');
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
  FOR I:=1 TO 12 DO
    BEGIN
      FR:=FAKR(I); (* AUFRUF DER FUNKTION FAKR *)
      FI:=FAK(I); (* AUFRUF DER FUNKTION FAK *)
      WRITELN(' ',I:3,' ',FR:15,' ',FI:15,' ');
    END;
  FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* FAKULTAE *)
```

```
*****
* REKURSIVE UND ITERATIVE FAKULTAETS-BERECHNUNG *
*****
* 1 * 1 * 1 *
* 2 * 2 * 2 *
* 3 * 6 * 6 *
* 4 * 24 * 24 *
* 5 * 120 * 120 *
* 6 * 720 * 720 *
* 7 * 5040 * 5040 *
* 8 * 40320 * 40320 *
* 9 * 362880 * 362880 *
* 10 * 3628800 * 3628800 *
* 11 * 39916800 * 39916800 *
* 12 * 479001600 * 479001600 *
*****
```

Übersicht über Standard-Prozeduren/Funktionen

STANDARD-PROZEDUREN/FUNKTIONEN SIND IN PASCAL VORDEFINIERT. SIE KÖNNEN SO GENUTZT WERDEN, WIE WENN SIE MIT EINEM GÜLTIGKEITSBEREICH VEREINBART WÜRDEN, DER DAS GANZE PROGRAMM UEBERDECKT. EINE ENTSPRECHENDE UEBERSICHT IST VON LOCHKARTEN EINZULESEN UND ZU DRUCKEN.

```
PROGRAM STANDARD(INPUT,OUTPUT);
(* STANDARD-PROZEDUREN/FUNKTIONEN-UEBERSICHT *)
(* PROZEDUR *)
VAR
  I,KARTEN,SPALTEN:INTEGER;
PROCEDURE KOPIE(KARTEN,SPALTEN:INTEGER);
VAR
```

```
*****
STANDARD-PROZEDUREN/FUNKTIONEN
*****
STANDARDPROZEDUREN:
GIBT SPEICHERPLATZ FREI, DER NICHT LAENGER
VON DYNAMISCHEN VARIABLEN BELEGT IST
UEBERTRAEGT EINE FILEKOMPONENTE AUF DIE
```

```

CH:CHAR;
I,J:INTEGER;
BEGIN (* INHALT VON LK WIRD GEDRUCKT *)
FOR I:=1 TO KARTEN DO
BEGIN
FOR J:=1 TO SPALTEN DO BEGIN
READ(CH);
WRITE(CH);
END; (* FOR J *)
READLN; (* ABSCHLUSS EINER LOCHKARTE *)
WRITELN (* DRUCK EINER ZEILE *)
END; (* FOR I *)
END; (* KOPIE *)
BEGIN
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
WRITELN('STANDARD-PROZEDUREN/FUNKTIONEN':40);
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
READLN(KARTEN,SPALTEN); (* LK-ANZAHL/SPALTEN *)
KOPIE(KARTEN,SPALTEN);
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
END. (* STANDARD *)

```

```

NEW: ENTSPRECHENDE PUFFERVARIABLE
SCHAFFT SPEICHERPLATZ FUER NEUE DYNAMI-
SCHE VARIABLEN
PACK: UEBERTRAEGT DATEN VON EINER NORMALEN AUF
EINE GEPACKTE DARSTELLUNG
PAGE: BEWIRKT UEBERGANG ZUR NAECHSTEN DRUCK-
SEITE
PUT: FUEGT DEN INHALT EINER PUFFERVARIABLEN
IHREM FILE HINZU
READ: WEIST DEN WERT EINER FILEKOMPONENTEN EINER
VARIABLEN ZU
READLN: FUEHRT READ AUS UND GEHT ZUR NAECHSTEN
ZEILE EINES TEXTFILES
RESET: EROEFFNET EIN FILE, UM VON IHM ZU LESEN
REWRITE: EROEFFNET EIN FILE, UM DARAUF ZU SCHREI-
BEN
UNPACK: UEBERTRAEGT DATEN VON EINER GEPACKTEN
AUF EINE NORMALE DARSTELLUNG
WRITE: WEIST EINEM FILE EINEN WERT ZU
WRITELN: FUEGT EINEM TEXTFILE 0,1 ODER VERSCHIE-
DENE ZEICHEN UND EINE ZEILENENDEMARKE
HINZU
#####
STANDARD FUNKTIONEN:
ABS: BERECHNET DEN ABSOLUTBETRAG DES AKTUEL-
LEN PARAMETERWERTES
ARCTAN: BERECHNET DEN ARKUSTANGENS DES AKTUELLEN
PARAMETERWERTES
CHR: BESTIMMT DAS ZEICHEN, DESSEN ORDNUNGS-
ZAHL GEGEBEN IST
COS: BERECHNET DEN COSINUS
EOF: TESTET, OB DAS ENDE EINES FILES ERREICHT
IST
EOLN: TESTET, OB DAS ENDE EINER ZEILE IN EINEM
TEXTFILE ERREICHT IST
EXP: BERECHNET DIE EXPONENTIALFUNKTION
LN: BERECHNET DEN NATUERLICHEN LOGARITHMUS
ODD: TESTET, OB EINE ZAHL UNGERADE IST
ORD: BESTIMMT DIE ORDNUNGSZAHL EINES WERTES
EINES GRUNTYPUS
PRED: BESTIMMT DEN VORGAENGER EINES WERTES EIN-
ES GRUNTYPUS
ROUND: RUNDET EINE REELLE ZAHL AUF DIE NAECHSTE
GANZE ZAHL
SIN: BERECHNET DEN SINUS
SQ: BERECHNET DAS QUADRAT
SQRT: BERECHNET DIE QUADRATWURZEL
SUCC: BESTIMMT DEN NACHFOLGER EINES WERTES EIN-
ES GRUNTYPUS
TRUNC: SCHNEIDET EINE REELLE ZAHL HINTER DEM
DEZIMALPUNKT AB
#####

```

Einführung der Vektorrechnung

ZUR UNTERSTUETZUNG DER ARBEIT MIT VEKTOREN SIND BESONDERS HAEUFIG BENUTZTE OPERATOREN FUER DREIDIMEN- SIONALE VEKTOREN BEFEITZUSTELLEN UND IHRE ANWENDUNG ZU DEMONSTRIEREN.

```

PROGRAM VEKTOREN(INPUT,OUTPUT);
(* VEKTOR-RECHNUNG *)
(* FUNKTION,PROZEDUR,FOR=ANWEISUNG *)
TYPE (* TYPDEFINITIONSTEIL *)
ACHSEN=(X,Y,Z); (* EXPLIZITER AUFZAHLUNGSTYP *)
VEKTOR=ARRAY[ACHSEN] OF REAL;
VAR
I,N:INTEGER; A,B,C:VEKTOR;
FUNCTION SKALARPR(A,B:VEKTOR):REAL;
VAR
S:REAL; R:ACHSEN;
BEGIN
S:=0;
FOR R:=X TO Z DO S:=S+A[R]*B[R];
SKALARPR:=S (* SKALARPRODUKT DER VEKTOREN A,B *)
END; (* SKALARPR *)
PROCEDURE VEKTSUMM(A,B:VEKTOR;VAR C:VEKTOR);
BEGIN
C[X]:=A[X]+B[X];
C[Y]:=A[Y]+B[Y];
C[Z]:=A[Z]+B[Z] (* SUMME DER VEKTOREN A,B *)
END; (* VEKTSUMM *)
PROCEDURE VEKTDIFF(A,B:VEKTOR;VAR C:VEKTOR);

```

```

*****
VEKTORRECHNUNG IM 3-DIM. RAUM
*****
1. BEISPIEL:
VEKTOR A=( 2,0000 0 0 )
VEKTOR B=( 0 2,0000 0 )
SUMME =( 2,0000 2,0000 0 )
DIFF =( 2,0000 -2,0000 0 )
PRÖD =( 0 0 4,0000 )
SKPRO = 0

2. BEISPIEL:
VEKTOR A=( 10,0000 10,0000 0 )
VEKTOR B=( 10,0000 10,0000 10,0000 )
SUMME =( 20,0000 20,0000 10,0000 )
DIFF =( 0 0 -10,0000 )
PRÖD =( 99,9999 -99,9999 0 )
SKPRO = 199,9996

3. BEISPIEL:
VEKTOR A=( 5,0000 0 5,0000 )
VEKTOR B=( -20,0000 -10,0000 5,0000 )

```

```

BEGIN
C[X]:=A[X]-B[X];
C[Y]:=A[Y]-B[Y];
C[Z]:=A[Z]-B[Z] (* DIFFERENZ DER VEKTOREN A,B *)
END; (* VEKTDIFF *)
PROCEDURE VEKTORPR(A,B:VEKTOR;VAR C:VEKTOR);
BEGIN
C[X]:=A[X]*B[Z]-A[Z]*B[X];
C[Y]:=A[Z]*B[X]-A[X]*B[Z];
C[Z]:=A[X]*B[Y]-A[Y]*B[X] (* VEKTORPRODUKT A,B *)
END; (* VEKTORPR *)
PROCEDURE LIESVEKT(VAR V:VEKTOR);
BEGIN
READLN(V[X],V[Y],V[Z]) (* LIES VEKTOR V *)
END; (* LIESVEKT *)
PROCEDURE DRUCKVEK(V:VEKTOR);
BEGIN
WRITELN('V[X]:',V[X];V[Y]:',V[Y];V[Z]:',V[Z]);
END; (* DRUCKVEK *)
BEGIN
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
WRITELN('VEKTORRECHNUNG IM 3-DIM. RAUM':40);
FOR I:=1 TO 50 DO WRITE('#'); WRITELN;
READLN(N); WRITELN('N':50);
FOR I:=1 TO N DO
BEGIN
WRITELN(I:39,'. BEISPIEL:');
LIESVEKT(A); WRITE('VEKTOR A='); DRUCKVEK(A);
LIESVEKT(B); WRITE('VEKTOR B='); DRUCKVEK(B);
WRITE('SUMME ='); VEKTSUMM(A,B,C); DRUCKVEK(C);
WRITE('DIFF ='); VEKTDIFF(A,B,C); DRUCKVEK(C);
WRITE('PRÖD ='); VEKTORPR(A,B,C); DRUCKVEK(C);
WRITELN('SKPRO =',SKALARPR(A,B):10:4);
WRITELN('N':50);
END; (* FOR I *)
END. (* VEKTOREN *)

```

```

SUMME =( -15,0000 -10,0000 10,0000 )
DIFF =( 25,0000 10,0000 0 )
PRÖD =( 49,9999 -124,9997 -49,9999 )
SKPRO = -74,9999

4. BEISPIEL:
VEKTOR A=( 15,0000 15,0000 15,0000 )
VEKTOR B=( -15,0000 -15,0000 15,0000 )
SUMME =( 0 0 30,0000 )
DIFF =( 30,0000 30,0000 0 )
PRÖD =( 449,9992 -449,9992 0 )
SKPRO = -224,9996

5. BEISPIEL:
VEKTOR A=( 7,5000 -7,5000 7,5000 )
VEKTOR B=( -7,5000 7,5000 7,5000 )
SUMME =( 0 0 15,0000 )
DIFF =( 15,0000 -15,0000 0 )
PRÖD =( -112,4997 -112,4997 0 )
SKPRO = -56,2499

```

Umkehrung einer Zeichenkette

DIE ARBEITSWEISE EINER REKURSIV ARBEITENDEN PROZEDUR IST AN HAND DER UMKEHRUNG EINER BELIBIG LANGEN ZEICHENKETTE, DIE DURCH EIN STERN-SYMBOL BEGRENZT WIRD, ZU ZEIGEN!

```

PROGRAM REKURSION(INPUT,OUTPUT);
(* SELBST-REKURSION *)
(* PROCEDURE,LOKAL *)
PROCEDURE REKUR; (* OHNE PARAMETER *)
VAR
CH:CHAR; (* LOKALE GROESZE *)
BEGIN (* REKURSIVE PROZEDUR *)
READ(CH); WRITE(CH);
IF CH = '*' THEN WRITELN;
IF NOT ( CH = '*' ) THEN REKUR;
WRITE(CH);
END; (* REKUR *)
BEGIN
WRITELN('UMKEHRUNG EINER ZEICHENKETTE':39);
WRITELN('=====','=====':39);
REKUR; (* PROZEDURAUFRUF *)
WRITELN;
WRITELN; WRITELN(' '); WRITELN('HINWEIS:');
WRITELN('EINE SELBST-REKURSION LIEGT DANN VOR',
', WENN EINE');
WRITELN('PROZEDUR/FUNKTION EINE ANWEISUNG ENTH',
'ALT, DIE AUF');
WRITELN('SIE SELBST VERWEIST,');
END. (* REKURSION *)

```

```

UMKEHRUNG EINER ZEICHENKETTE
=====
N E G E R L E G U K K U G E L R E G E N
* N E G E R L E G U K K U G E L R E G E N
HINWEIS:
EINE SELBST-REKURSION LIEGT DANN VOR, WENN EINE
PROZEDUR/FUNKTION EINE ANWEISUNG ENTHAELT, DIE AUF
SIE SELBST VERWEIST.

```

PASCAL-Standardbezeichner

DIE 40 PASCAL-STANDARDBEZEICHNER SIND VON LOCHKARTEN EINZULESEN UND IN FUENF SPALTEN ZU DRUCKEN. AUF DEN LOCHKARTEN SIND SIE DURCH EIN ODER MEHRERE LEERZEICHEN GETRENNT.

```

PROGRAM ALFAWORD(INPUT,OUTPUT);
(* EIN-U. AUSGABE VON WERTEN VON ALFA-GROESZEN *)
(* ALFATYP,READLN,PROZEDUR,LOKAL *)
VAR
A:ALFA;
I,J:INTEGER;

```

```

*****
ABS ARCTAN BOOLEAN CHAR CHR
COS DISPOSE EOF EOLN EXP
FALSE GET INTEGER INPUT LN
MAXINT NEW ODD URD OUTPUT
PACK PAGE PRED PUT READ

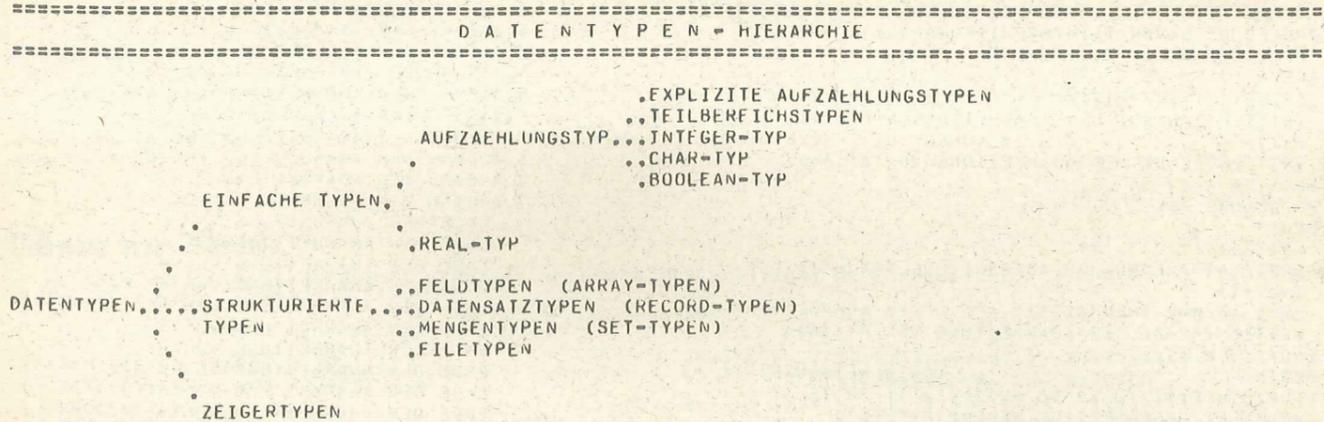
```



```

VAR
I:INTEGER;
PROCEDURE COPYCARD;
VAR
CH:CHAR;
BEGIN
WHILE NOT EOF DO
BEGIN
WHILE NOT EOLN DO BEGIN READ(CH);
WRITE(CH)
END;
READLN;
END;
(* COPYCARD KOPIERT DEN INHALT BELIEBIG VIELER
LUCHKARTEN ORIGINALGETREU AUF EINE DRUCKLISTE. *)
BEGIN
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
WRITELN('D A T E N I Y P E N = H I E R A R C H I E : 6 7');
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
COPYCARD; (* PROZEDUR-AUFRUF *)
(* DURCH DIESEN PROZEDUR-AUFRUF WERDEN ALLE NOCH
ANLIEGENDEN LUCHKARTEN EINGELESEN UND IHRE INHAL-
TE GEDRUCKT. *)
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
END. (* DATENTYP *)

```



VORDEFINIERTE STANDARDTYPEN SIND:
INTEGER, BOOLEAN, CHAR, REAL UND
FILE TEXT.

Komplexe Arithmetik

DURCH DIE EINFUEHRUNG VON PROZEDUREN IST DIE ARBEIT MIT KOMPLEXEN GROESZEN ZU RATIONALISIEREN. DIE ANWENDUNG DER PROZEDUREN IST ZU DEMONSTRIEREN.

```

PROGRAM KOMPLEX(INPUT,OUTPUT);
(* ARITHMETIK KOMPLEXER ZAHLEN *)
(* ARRAY, RECORD, BOOLEAN, PROCEDURE *)
TYPE
KZAHL=RECORD
R,J:REAL
END; (* TYP EINER KOMPLEXEN ZAHL *)
VAR
I:INTEGER; A,B,C:REAL; L:BOOLEAN;
U,V,W,X,Y,Z:KZAHL;
F:ARRAY[1..10] OF KZAHL;
PROCEDURE ADD(F,G:KZAHL; VAR H:KZAHL (* SUMME *));
BEGIN
H.R:=F.R + G.R;
H.J:=F.J + G.J
END; (* ADD *)
PROCEDURE SUB(F,G:KZAHL; VAR H:KZAHL (* DIFF. *));
BEGIN
H.R:=F.R - G.R;
H.J:=F.J - G.J
END; (* SUB *)
PROCEDURE MUL(F,G:KZAHL; VAR H:KZAHL (* PROD. *));
BEGIN
H.R:=F.R*G.R - F.J*G.J;
H.J:=F.R*G.J + F.J*G.R
END; (* MUL *)
PROCEDURE DIVI(F,G:KZAHL; VAR H:KZAHL (* QUOT. *));
VAR
N:REAL;
BEGIN
N:=G.R*G.R + G.J*G.J;
H.R:=(F.R*G.R + F.J*G.J)/N;
H.J:=(F.J*G.R - F.R*G.J)/N
END; (* DIVI *)
END; (* ASS *)
PROCEDURE COM(F,G:KZAHL; VAR B:BOOLEAN);
BEGIN
(* VERGLEICH ZWEIER KOMPLEXER ZAHLEN *)
IF (F.R=G.R) AND (F.J=G.J) THEN B:=TRUE
ELSE B:=FALSE
END; (* COM *)
PROCEDURE LIES (VAR F:KZAHL);
BEGIN
(* EINLESEN EINES WERTEPAARES *)
READ(F,R);
READ(F,J)
END; (* LIES *)
PROCEDURE NULL (VAR F:KZAHL);
BEGIN
(* ZUWEISUNG DES WERTES NULL *)
F.R:=0;
F.J:=0
END; (* NULL *)
PROCEDURE PRINT(F:KZAHL);
BEGIN
(* DRUCK EINER KOMPLEXEN ZAHL *)
IF F.J>=0 THEN WRITE(F.R:9:2, ' + ', F.J:9:2, ' * I ')
ELSE WRITE(F.R:9:2, ' - ', F.J:9:2, ' * I ')
END; (* PRINT *)
PROCEDURE ABSO(F:KZAHL; VAR B:REAL (* BETRAG *));
BEGIN
B:=SQRT(SQR(F.R) + SQR(F.J))
END; (* ABSO *)
FOR I:=1 TO 103 DO WRITE('# '); WRITELN;
WRITELN('RECHNUNG MIT KOMPLEXEN ZAHLEN:67');
FOR I:=1 TO 103 DO WRITE('# '); WRITELN;
WRITELN('KOMPL.ZAHL:22, 'KONJ.KOMPL.ZAHL:22,
'SUMME:22, 'PRODUKT:22, ' BETRAG D.ZAHL:15);
NULL(U);
FOR I:=1 TO 10 DO

```

```

END; (* DIVI *)
PROCEDURE KON(F:KZAHL; VAR G:KZAHL (* KONJ.Z. *));
BEGIN
(* KONJUGIERT KOMPLEXE ZAHL *)
G.R:=F.R;
G.J:=-F.J
END; (* KON *)
PROCEDURE ASS(F:KZAHL; VAR G:KZAHL);
BEGIN
(* WERTWEISUNG *)
G.R:=F.R;
G.J:=F.J

```

```

BEGIN
LIES(F[I]); PRINT(F[I]);
KON(F[I],V); PRINT(V);
ADD(U,F[I],U); PRINT(U);
MUL(F[I],U,W); PRINT(W);
ABSO(F[I],A); WRITE(A);
WRITELN
END; (* FOR I *)
END. (* KOMPLEX *)

```

RECHNUNG MIT KOMPLEXEN ZAHLEN

KOMPL.ZAHL	KONJ.KOMPL.ZAHL	SUMME	PRODUKT	BETRAG D.ZAHL
100,00 + 100,00*I	100,00 - 100,00*I	100,00 + 100,00*I	0 + 19999,96*I	1,41421E+02
3,00 + 4,00*I	3,00 + 4,00*I	103,00 + 96,00*I	693,00 - 124,00*I	5,00000E+00
1,00 + 0*I	1,00 + 0*I	104,00 + 96,00*I	104,00 + 96,00*I	0,99999E+00
0 + 0*I	0 + 0*I	104,00 + 96,00*I	0 + 0*I	0
0,05 + 0,00*I	0,05 + 0,00*I	104,05 + 96,00*I	5,30 + 4,70*I	5,00100E-02
-4,50 + 5,40*I	-4,50 + 5,40*I	99,55 + 90,60*I	41,26 - 945,26*I	7,02922E+00
3,14 + 3,14*I	3,14 + 3,14*I	102,69 + 87,46*I	597,35 - 47,86*I	4,44275E+00
25,00 + 109,00*I	25,00 - 109,00*I	127,69 + 196,46*I	-18221,56 + 18829,78*I	1,11830E+02
5,50 + 6,60*I	5,50 - 6,60*I	133,19 + 203,06*I	-607,63 + 1995,88*I	8,59127E+00
-0,01 + 0,05*I	-0,01 - 0,05*I	133,18 + 203,11*I	-11,49 + 4,63*I	5,09902E-02

Iterative Berechnung des arithmetischen Mittels

IN DER MESZTECHNIK WIRD OFT DAS ARITHMETISCHE MITTEL EINER FOLGE VON MESZWERTEN BENUTZT. DIESE BERECHNUNG SOLL ITERATIV ERFOLGEN, DAMIT JEDERZEIT WEITERE MESZWERTE BERUECKSICHTIGT WERDEN KÖNNEN.

```

PROGRAM MITTEL(INPUT,OUTPUT);
(* ITERATIVE MITTELWERTBERECHNUNG *)
(* PROZEDUR, STANDARDFILE, WHILE-ANWEISUNG *)
VAR
I:INTEGER;
ZM,ZI:REAL;
PROCEDURE MITTELWERT(VAR Z:REAL; ZI:REAL;
VAR I:INTEGER);
BEGIN
Z:=(I*Z+ZI)/(I+1); (* NEUER MITTELWERT *)
I:=I+1 (* ERHOEHUNG DES ZAEHLERS DER MESZDATEN *)
END; (* MITTELWERT *)
BEGIN
WRITE('ITERATIVE BERECHNUNG DES ARITHMETISCHEN');
WRITELN(' MITTELWER=');
WRITE('TES EINER BELIEBIGEN ANZAHL VON MESZDAT');
WRITELN('EN, DIE EIN=');
WRITELN('GELESEN WERDEN:');
WRITELN(' ');
WRITELN('MESZWERT:15, 'MITTELWERT:15, ' ANZAHL');
I:=0; (* ANFANGSWERTZUWEISUNG AN ANZAHL DATEN *)
ZM:=0; (* ANFANGSWERTZUWEISUNG AN MITTELWERT *)
WHILE NOT EOF DO
BEGIN
READLN(ZI); (* LIES NAECHSTEN MESZWERT *)
MITTELWERT(ZM,ZI,I); (* PROZEDURAUFRUF *)
WRITELN(ZI,ZM,I:7)
END;
FOR I:=1 TO 50 DO WRITE(' '); WRITELN;
END. (* MITTEL *)

```

ITERATIVE BERECHNUNG DES ARITHMETISCHEN MITTELWERTES EINER BELIEBIGEN ANZAHL VON MESZDATEN, DIE EINGELESEN WERDEN:

MESZWERT	MITTELWERT	ANZAHL
0,99999E+00	0,99999E+00	1
2,00000E+00	1,50000E+00	2
3,00000E+00	2,00000E+00	3
4,00000E+00	2,50000E+00	4
5,00000E+00	3,00000E+00	5
5,99999E+00	3,50000E+00	6
6,99999E+00	3,99999E+00	7
8,99999E+00	4,62499E+00	8
9,99999E+00	5,22221E+00	9
1,10000E+01	5,79999E+00	10
7,99999E+00	5,99998E+00	11
1,55000E+01	6,79165E+00	12
2,63500E+01	8,29614E+00	13
1,21755E+02	1,64003E+01	14
-1,21755E+02	7,18998E+00	15
-2,63500E+01	5,09373E+00	16
-1,55000E+01	3,88233E+00	17
0	3,66665E+00	18
9,99500E+02	5,60789E+01	19
-5,55499E+02	2,55000E+01	20
-4,44000E+02	3,14280E+00	21

Teilnehmerliste

DIE HOERER EINER VORLESUNG SIND IN EINER LISTE NAMENTLICH ZU ERFASSEN. AUSZERDEM SIND DIE TAETIGKEIT, DAS ALTER UND DIE SEKTIONSNUMMER ANZUGEHEN. DIE NAMEN DER JUENGSTEN HUERER SIND GESUCHT.

```

PROGRAM HOERER(INPUT,OUTPUT);
(* ARBEIT MIT EINER TEILNEHMER-LISTE *)
(* DATENSATZTYPEN, FILE, PROZEDUR *)
CONST
MAX=250; (* MAXIMALE LISTENLAENGE *)
NL=15;
TYPE
NAME=PACKED ARRAY[1..NL] OF CHAR;
HOERER=RECORD (* DATENSATZTYP *)
NR:0..MAX;

```

```

WRITELN('LISTE DER HOERER:59); WRITELN(':103);
WRITELN('LAUF.NR:10, 'FAMILIENNAME:NL,
'VORNAME:NL+1, 'TAETIGKEIT:NL+2, 'ALTER:10,
'ANMERKUNG:29);
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
I:=0;
WHILE NOT EOF(INPUT) DO
BEGIN
I:=I+1; (* ZAEHLER DER TEILNEHMER *)
TW[I].NR:=I;

```

```

ZUNAME:NAME;
VORNAME:NAME;
TAETIGKEIT:NAME;
ALTER:10..65;
SEKTION:0..15
END; (* HOERER *)

VAR
I,ANZ,JUNG:INTEGER; TEILN:HOERER;
TN:ARRAY[1..MAX] OF HOERER;
PROCEDURE LIESNAME(VAR N:NAME);
VAR
J:INTEGER;
BEGIN
(* LIEST STRING DER LAENGE NL *)
FOR J:=1 TO NL DO READ(N[J]);
END; (* LIESNAME *)
PROCEDURE DRUKNAME(N:NAME);
VAR
J:INTEGER;
BEGIN
(* DRUCKT STRING DER LAENGE NL *)
FOR J:=1 TO NL DO WRITE(N[J]);
END; (* DRUKNAME *)
BEGIN
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
WRITELN('VORLESUNG: INFORMATIK'); WRITELN(' ');

LIESNAME(TN[I],ZUNAME);
LIESNAME(TN[I],VORNAME);
LIESNAME(TN[I],TAETIGKEIT);
READLN(TN[I],ALTER,TN[I],SEKTION);
WRITE(I:10,' '); (* ANFANG EINER DRUCKZEILE *)
DRUKNAME(TN[I],ZUNAME); WRITE(' ');
DRUKNAME(TN[I],VORNAME); WRITE(' ');
DRUKNAME(TN[I],TAETIGKEIT); WRITE(' ');
WRITE(TN[I].ALTER:2,' JAHRE, SEKTION: ');
IF TN[I].SEKTION=0 THEN WRITELN('KEINE') ELSE
WRITELN(TN[I].SEKTION:2);
END; (* OF WHILE *) WRITELN(' ':103);
ANZ:=I; (* ANZAHL DER TEILNEHMER *)
WRITELN('TEILNEHMER INSGESAMT:',ANZ:3);
JUNG:=65;
FOR I:=1 TO ANZ DO
IF TN[I].ALTER < JUNG THEN JUNG:=TN[I].ALTER;
WRITE('JUENGSTE HOERER DER VORLESUNG: ');
FOR I:=1 TO ANZ DO
IF TN[I].ALTER=JUNG THEN
BEGIN DRUKNAME(TN[I],ZUNAME); WRITE(' ') END;
WRITELN;
FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
END. (* HOERER *)

```

VORLESUNG: INFORMATIK

LISTE DER HOERER

LAUF.NR	FAMILIENNAME	VORNAME	TAETIGKEIT	ALTER	ANMERKUNG
1	GUTENBERGER	HANS-ULLRICH	KONSTRUKTEUR	27 JAHRE, SEKTION: 6	
2	LAMPENSCHAEKF	HEINZ-JUERGEN	TECHNOLOGE	28 JAHRE, SEKTION: KEINE	
3	LICHTENBERGER	KARL-HEINZ	MATHEMATIKER	29 JAHRE, SEKTION: 4	
4	APPELRATH	KLAUS-ULLRICH	CHEMIKER	27 JAHRE, SEKTION: KEINE	
5	SCHNEIDERREITH	HANS-JUERGEN	INFORMATIKER	30 JAHRE, SEKTION: 3	
6	SEMMELEWISZ	ANNE-ROSE	CHEMIKERIN	26 JAHRE, SEKTION: 8	
7	WOLLSCHLEGEL	ROSEMARIE	STUDENTIN	20 JAHRE, SEKTION: 7	
8	GASSENBRENNER	HEINZ-JUERGEN	STUDENT	24 JAHRE, SEKTION: 9	
9	ROHDENSTOCK	ADELHEID	STUDENTIN	20 JAHRE, SEKTION: 5	
10	STRUMBERGER	BARBARA	PHYSIKERIN	25 JAHRE, SEKTION: 4	

TEILNEHMER INSGESAMT: 10
JUENGSTE HOERER DER VORLESUNG: WOLLSCHLEGEL, ROHDENSTOCK

Mitgliederverzeichnis

DIE MITGLIEDER EINER INSTITUTION SIND NAMENTLICH AUFZULISTEN. AUSZERDEM SIND DIE NAMEN ANZUGEBEN, DIE AUS DIESER LISTE ZU STREICHEN SIND.

```

PROGRAM KARTEI(INPUT,OUTPUT);
(* PERSONEN=DATEI *)
(* DATENSATZTYPEN,WITH=ANWEISUNG *)
CONST
NL=13; ANZ=20;
TYPE
C:CHAR; (* TRUE,FALSE DURCH T,F VERSCHLUESSELT *)
WORT=ARRAY[1..NL] OF CHAR;
DATUM=RECORD TAG:1..31;
MONAT:1..12;
JAHR:1800..2000 END; (* DATUM *)
PERSON=RECORD VORNAME:ALFA;
NAME:WORT;
PNR:INTEGER;
GEBURT:DATUM;
SEX:(M,W);
LEBEND:BOOLEAN END; (* PERSON *)
VAR
I,J,K:INTEGER; X:ARRAY[1..ANZ] OF PERSON;
PROCEDURE READBOOL(VAR X:BOOLEAN);
VAR
C:CHAR; (* TRUE,FALSE DURCH T,F VERSCHLUESSELT *)
BEGIN
(* LIES GROESZEN VOM TYP BOOLEAN *)
REPEAT READ(C) (* LEERZEICHEN UEBERLESEN *)
UNTIL (C='T') OR (C='F');
IF C='T' THEN X:=TRUE ELSE X:=FALSE;
END; (* READBOOL *)
PROCEDURE LIESPERS(VAR Y:PERSON);
PROCEDURE DRUCKPER(Y:PERSON);
VAR
I:INTEGER;
BEGIN
WITH Y DO
BEGIN
FOR I:=1 TO NL DO WRITE(NAME[I]);
WRITE(VORNAME:9);
WRITE(PNR:10);
WRITE(GEBURT,TAG:5,GEBURT.MONAT:3,GEBURT.JAHR:5
(* GEBURTSDATUM *));
IF SEX='M' THEN WRITE(' MAENNL. ');
ELSE WRITE(' WEIBL. ');
IF LEBEND THEN WRITE(' LEBEND ');
ELSE WRITE(' T U I ');
END; (* WITH *)
END; (* DRUCKPER *)
BEGIN
FOR I:=1 TO 103 DO WRITE('*'); WRITELN;
WRITELN('MITGLIEDER = DATEI':60);
FOR I:=1 TO 103 DO WRITE('*'); WRITELN;
WRITELN('LAUF.NR. ':12,'ZUNAME ':14,'VORNAME ':10,
'MITGL.NR. ':10,'GEBURTS DAT. ':13,'GESCHL. ':8,
'AKTIVITAE ':12,'ANMERKUNGEN ':20);
FOR I:=1 TO 103 DO WRITE('*'); WRITELN;
K:=1; LIESPERS(X[K]);
WHILE X[K].NAME[I] <> '*' DO

```

```

VAR
I:INTEGER; C:CHAR;
BEGIN
REPEAT READ(Y.NAME[I]) UNTIL Y.NAME[I] <> ' ';
IF Y.NAME[I] <> '*' THEN
BEGIN
FOR I:=2 TO NL DO READ(Y.NAME[I]);
FOR I:=1 TO 8 DO READ(Y.VORNAME[I]);
READ(Y.PNR);
READ(Y.GEBURT.TAG,Y.GEBURT.MONAT,Y.GEBURT.JAHR)
(* LIES GEBURTS DATUM *);
REPEAT READ(C) UNTIL (C='M') OR (C='W');
IF C='M' THEN Y.SEX:=M ELSE Y.SEX:=W;
READBOOL(Y.LEBEND);
END; (* IF *)
END; (* LIESPERS *)
BEGIN
WRITE(K:12,' ':2);
DRUCKPER(X[K]); WRITELN;
K:=K+1;
LIESPERS(X[K]);
END; (* WHILE *)
WRITE('AUSGESCHIEDENE MITGLIEDER SIND: ');
FOR I:=1 TO K-1 DO
IF NOT X[I].LEBEND THEN
BEGIN
FOR J:=1 TO NL DO WRITE(X[I].NAME[J]);
WRITE(' ');
END; (* IF *) WRITELN;
FOR I:=1 TO 103 DO WRITE('*'); WRITELN;
END. (* KARTEI *)

```

***** MITGLIEDER = DATEI *****

LAUF.NR.	ZUNAME	VORNAME	MITGL.NR.	GEBURTS DAT.	GESCHL.	AKTIVITAE	ANMERKUNGEN
1	WEISZENBORN	GUENTHER	50403140	1 1 1930	MAENNL.	LEBEND	
2	DOSTOJEWSKI	TAMARA	10982356	10 10 1927	WEIBL.	LEBEND	
3	STEINBERG	ROSWITA	80324156	12 12 1910	WEIBL.	LEBEND	
4	DICKENS	CHARLIE	14050127	12 12 1860	MAENNL.	T O T	
5	SEIDELBAST	KLAUS	21334556	4 10 1960	MAENNL.	LEBEND	
6	KLOPSTOCK	MARTINA	44257802	19 1 1923	WEIBL.	T O T	
7	WAGENTHIN	WERNER	98764523	9 12 1945	MAENNL.	LEBEND	
8	SCHNEIDEROTH	JUERGEN	98324501	9 8 1930	MAENNL.	LEBEND	

AUSGESCHIEDENE MITGLIEDER SIND: DICKENS, KLOPSTOCK

Künstlerdatei

FUER EINE KUENSTLERAGENTUR IST EINE DATEI AUFZUSTELLEN, DIE ALLE WESENTLICHEN ANGABEN UEBER DIE KUENSTLER, DIE DURCH DIE AGENTUR VERMITTELT WERDEN, ENTHAELT.

```

PROGRAM KUENSTE(INPUT,OUTPUT);
(* VARIABLE SAETZE VON KUENSTLERDATEN *)
(* DATENSATZTYPEN,CASE=ANWEISUNG,VARIANTENTEIL *)
TYPE
KUENSTLERTYP=(SCHAUSPIELER,SAENGER);
ROTYP=(KOMISCH,DRAMATISCH,ROMANTISCH,SATIRISCH);
GESCHLECHT=(M,W);
ROLLENTYP=SET OF ROTYP;
STIMMTYP=(SOPRAN,ALT,TENOR,BASS);
DARSTELLER=RECORD
NAME:ALFA;
ALTER:8..90;
CASE KUNST:KUENSTLERTYP OF
SCHAUSPIELER:(GESCHL:GESCHLECHT;
ROLLE:ROLLENTYP);
SAENGER:(STIMME:STIMMTYP)
END; (* DARSTELLER *)
VAR
I,J:INTEGER; ROLLINDEX:ROTYP;
DARST:DARSTELLER;
DARDAT:ARRAY[1..100] OF DARSTELLER;
BEGIN
FOR I:=1 TO 103 DO WRITE('*'); WRITELN;
WRITELN('DATEI DER KUENSTLER=AGENTUR "G U T E ',
'LAUNE ZU ALLERZEIT');
FOR I:=1 TO 103 DO WRITE('*'); WRITELN;
WITH DARST DO
BEGIN
NAME:='MAYERLOH';
ALTER:=35;
KUNST:=SCHAUSPIELER;
GESCHL:=W;
ROLLE:=(KOMISCH,ROMANTISCH,SATIRISCH)
END; (* WITH *)
DARDAT[I]:=DARST; (* WERTZUWEISUNG AN 1.SATZ *)
WITH DARST DO
BEGIN
NAME:='ROTENBER';
KUNST:=SAENGER;
STIMME:=ALT
END; (* WITH *)
KUNST:=SCHAUSPIELER;
GESCHL:=M;
ROLLE:=(KOMISCH,ROMANTISCH,DRAMATISCH)
END; (* WITH *)
DARDAT[I]:=DARST; (* WERTZUWEISUNG AN 3.SATZ *)
FOR I:=1 TO 3 DO
BEGIN
WITH DARDAT[I] DO
BEGIN
WRITE(I:3,'. NAME: ',NAME,' ALTER: ',ALTER:2,
', JAHRE, KUNSTGATTUNG: ');
CASE KUNST OF
SCHAUSPIELER:
BEGIN
WRITE('SCHAUSPIELER ');
CASE GESCHL OF
W:WRITE(' GESCHLECHT: WEIBLICH ');
M:WRITE(' GESCHLECHT: MAENNLICH ');
END; (* CASE *)
WRITELN; WRITE(' ');
WRITE('ROLLE: ');
FOR ROLLINDEX:=KOMISCH TO SATIRISCH DO
IF ROLLINDEX IN ROLLE THEN
CASE ROLLINDEX OF
KOMISCH :WRITE('KOMISCHE, ');
DRAMATISCH:WRITE('DRAMATISCHE, ');
ROMANTISCH:WRITE('ROMANTISCHE, ');
SATIRISCH :WRITE('SATIRISCHE, ');
END; (* CASE *)
END; (* SCHAUSPIELER *)
SAENGER:
BEGIN
WRITE('SAENGER ');
CASE STIMME OF
SOPRAN:WRITE('SOPRAN');
ALT :WRITE('ALT');
TENOR :WRITE('TENOR');
BASS :WRITE('BASS');
END; (* CASE *)
END; (* SAENGER *)
END; (* CASE KUNST *)

```

```

DARDAT[2]:=DARST; (* WERTZUWEISUNG AN 2.SATZ *)
WITH DARST DO
BEGIN
  NAME:='HUNDEROT';
  ALTER:=28;
  WKITELN;
  END; (* WITH DARDAT[1] *)
  END; (* FOR I *)
  FOR I:=1 TO 103 DO WRITE('*'); WKITELN;
  END. (* KUENSTE *)

```

```

*****
DATEI DER KUENSTLER-AGENTUR "G U T E L A U N E Z U A L L E R Z E I T"
*****
1. NAME: MAYERLOH ALTER: 35 JAHRE, KUNSTGATTUNG: SCHAUSPIELER GESCHLECHT: WEIBLICH
   ROLLEN: KOMISCHE, ROMANTISCHE, SATIRISCHE,
2. NAME: ROTENBER ALTER: 35 JAHRE, KUNSTGATTUNG: SAENGER ALT
3. NAME: HUNDEROT ALTER: 28 JAHRE, KUNSTGATTUNG: SCHAUSPIELER GESCHLECHT: MAENNLICH
   ROLLEN: KOMISCHE, DRAMATISCHE, ROMANTISCHE,
*****

```

Kurbelwellen

DIE VOLUMINA VON KURBELWELLEN, DIE AUS ZYLINDERN UNTERSCHIEDLICHER RADII BESTEHEN UND BEIDSEITIG DURCH JE EINE VON ZWEI GLEICHEN HALBKUGELN BEGRENZT WERDEN, SIND ZU BERECHNEN.

```

PROGRAM VOLUMINA(INPUT,OUTPUT);
(* VOLUMEN VON KUGEL,ZYLINDER *)
(* DATENSATZTYPEN,PROZEDUR,MARKEN *)
TYPE
  GESTALT=(KUGEL,ZYLINDER);
  INHALTE=RECORD
    CASE ETIKETT:GESTALT OF
      KUGEL:(RADIUS:REAL);
      ZYLINDER:(RAD,HOEHE:REAL)
    END;
VAR
  I,J,ANZ1,ANZ2:INTEGER; VOL,GVOL:REAL;
  INH:INHALTE;
PROCEDURE LIESGEST(VAR G:GESTALT);
VAR
  I:1..5; EING:PACKED ARRAY[1..5] OF CHAR;
BEGIN
  FOR I:=1 TO 5 DO READ(EING[I]); READLN;
  IF EING='KUGEL' THEN G:=KUGEL ELSE
  IF EING='ZYLIN' THEN G:=ZYLINDER
  ELSE WRITELN('DATENFEHLER ! ');
END; (* LIESGEST *)
FUNCTION VOLUM(INH:INHALTE):REAL;
CONST
  PI=3.14159;
BEGIN
  WITH INH DO
    CASE ETIKETT OF
      KUGEL:VOLUM:=PI*SQR(RADIUS)*RADIUS*4.0/3.0;
      ZYLINDER:VOLUM:=PI*SQR(RAD)*HOEHE
    END
  END; (* VOLUM *)
BEGIN
  FOR I:=1 TO 103 DO WRITE('='); WKITELN;
  WRITELN('VOLUMINA-BERECHNUNG';61);
  FOR I:=1 TO 103 DO WRITE('='); WKITELN;

```

```

  READLN(ANZ1);
  FOR I:=1 TO ANZ1 DO
    BEGIN
      GVOL:=0;
      WRITELN(I:2,' KURBELWELLE');
      READLN(ANZ2);
      FOR J:=1 TO ANZ2 DO
        BEGIN
          WITH INH DO
            BEGIN
              LIESGEST(ETIKETT);
              CASE ETIKETT OF
                KUGEL:
                  BEGIN READLN(RADIUS);
                     WRITE('KUGEL: ');
                     WRITE(' RADIUS=',RADIUS:8:3)
                  END; (* KUGEL *)
                ZYLINDER:
                  BEGIN READLN(RAD,HOEHE);
                     WRITE('ZYLINDER:');
                     WRITE(' RADIUS=',RAD:8:3);
                     WRITE(' LAENGE=',HOEHE:8:3)
                  END (* ZYLINDER *)
              END; (* CASE *)
            END; (* WITH *)
          VOL:=VOLUM(INH);
          GVOL:=GVOL + VOL;
          WRITE(' DAS VOLUMEN IST:',VOL:8:2,' M**M**');
          WKITELN;
        END; (* FOR J *)
      WRITE('DAS VOLUMEN DER KURBELWELLE IST ':89);
      WRITELN(GVOL:8:2,' M**M**');
    END; (* FOR I *)
  FOR I:=1 TO 103 DO WRITE('='); WKITELN;
  END. (* VOLUMINA *)

```

VOLUMINA-BERECHNUNG

```

=====
1. KURBELWELLE
KUGEL: RADIUS= 0.350 DAS VOLUMEN IST: 0.18 M**M**
ZYLINDER: RADIUS= 0.350 LAENGE= 0.600 DAS VOLUMEN IST: 0.23 M**M**
ZYLINDER: RADIUS= 0.300 LAENGE= 0.050 DAS VOLUMEN IST: 0.01 M**M**
ZYLINDER: RADIUS= 0.250 LAENGE= 0.100 DAS VOLUMEN IST: 0.02 M**M**
ZYLINDER: RADIUS= 0.300 LAENGE= 0.050 DAS VOLUMEN IST: 0.01 M**M**
ZYLINDER: RADIUS= 0.250 LAENGE= 0.100 DAS VOLUMEN IST: 0.02 M**M**
ZYLINDER: RADIUS= 0.350 LAENGE= 0.600 DAS VOLUMEN IST: 0.23 M**M**
          DAS VOLUMEN DER KURBELWELLE IST 0.71 M**M**

2. KURBELWELLE
KUGEL: RADIUS= 0.500 DAS VOLUMEN IST: 0.52 M**M**
ZYLINDER: RADIUS= 0.500 LAENGE= 0.950 DAS VOLUMEN IST: 0.75 M**M**
ZYLINDER: RADIUS= 0.300 LAENGE= 0.100 DAS VOLUMEN IST: 0.03 M**M**
ZYLINDER: RADIUS= 0.500 LAENGE= 0.950 DAS VOLUMEN IST: 0.75 M**M**
          DAS VOLUMEN DER KURBELWELLE IST 2.04 M**M**
=====

```

Postleitzahlen

FUER EINE ANALYSE DER BEZIEHUNGEN ZU DEN LIEFERANTEN EINES BETRIEBES SIND AUS DER DATEI DER ANSCHRIFTEN ALLE POSTLEITZAHLEN ZU ERZEUGEN UND MIT EINER LAUFENDEN NUMMER VERSEHEN AUFZULISTEN.

```

PROGRAM PLZ(AHL(INPUT,OUTPUT,LIEFT));
(* ERZEUGUNG VON POSTLEITZAHLEN AUS TEXTFILE *)
(* FILETYPEN,PUFFERVARIABLE,GET,PUT *)
TYPE
  PLZT=1000..9999;
VAR
  I,J:INTEGER; LZ:BOOLEAN; PLZ:PLZT;
  LIEFT:TEXT; (* LIEFERANTENANSCHRIFTEN *)
PROCEDURE KOPIE(VAR F,G:TEXT);
VAR
  C:CHAR;
BEGIN (* UEBERTRAGUNG EINER ZEILE VON F AUF G *)
  WHILE NOT EOF(F) DO
    BEGIN
      WHILE NOT EOLN(F) DO
        BEGIN READ(F,C); WRITE(G,C) END;
        READLN(F); WKITELN(G);
      END (* WHILE *)
    END; (* KOPIE *)
  PROCEDURE LIESPZ(VAR F:TEXT;VAR X:PLZT;
                  VAR L:BOOLEAN);
    VAR
      I,J:INTEGER; C:CHAR;
      ZEI:PACKED ARRAY[CHAR] OF BOOLEAN;
    BEGIN (* BERECHNUNG DER PLZ *)
      FOR C:=CHR(0) TO CHR(255) DO ZEI[C]:=FALSE;
      FOR C:='0' TO '9' DO ZEI[C]:=TRUE;
      L:=FALSE; (* L INFORMIERT, OB PLZ ERRECHNET W. *)
      WHILE (NOT EOLN(F)) AND (NOT ZEI[F@]) DO GET(F);
      I:=0;
      WHILE (NOT EOLN(F)) AND ZEI[F@] DO
        BEGIN (* BERECHNUNG EINER GANZEN ZAHL *)
          L:=TRUE;
          J:=ORD(F@)-ORD('0');
          I:=10*I+J; (* HORNER-SCHEMA *)
          GET(F) (* N. ZEICHEN AN PUFFERVARIABLE F@ *)
        END; (* WHILE *)
      IF (I>999) AND (I<10000) THEN X:=I (* PLZ WURDE
      KORREKT ERRECHNET *) ELSE L:=FALSE;
      END; (* LIESPZ *)
    BEGIN
      FOR I:=1 TO 103 DO WRITE('#'); WKITELN;
      WKITELN('LIEFERANTEN-DATEI:');
      REWRITE(LIEFT);
      KOPIE(INPUT,LIEFT);
      RESET(LIEFT);
      KOPIE(LIEFT,OUTPUT);
      RESET(LIEFT); (* AKT.FILEPOSITION AN ANFANG *)
      WKITELN('=:53);
      WKITELN('DIE POSTLEITZAHLEN DER LIEFERANTEN:');
      J:=0; (* ANFANGSWERT AN ZAEHLER D.PLZ *)
      WHILE NOT EOF(LIEFT) DO
        BEGIN
          WHILE NOT EOLN(LIEFT) DO
            BEGIN
              LIESPZ(LIEFT,PLZ,LZ);
              IF LZ THEN BEGIN J:=J+1; WRITE(J:4,PLZ:5)
                END; (* PLZ WIRD GEDRUCKT *)
            END; (* WHILE *)
          IF J MOD 11=0 THEN WKITELN; (* NEUE ZEILE *)
          READLN(LIEFT);
          END; (* WHILE *)
        WKITELN;
        FOR I:=1 TO 103 DO WRITE('#'); WKITELN;
        WKITELN('POSTLEITZAHLEN AUS LIEFERANTEN-DATEI:65,
          ' ERZEUGEN');
        FOR I:=1 TO 103 DO WRITE('#'); WKITELN;
        END. (* PLZ(AHL *)

```

```

#####
LIEFERANTEN-DATEI:
VEB SCHWERMASCHINENBAU-KOMBINAT "ERNST THAELMANN" 3011 MAGDEBURG, MARIENSTR. 20
VEB SCHWERMASCHINENBAUKOMBINAT TAKRAF 7010 LEIPZIG, SEEBURGSTRASSE 14/20
VEB MASCHINENBAUHANDEL 3018 MAGDEBURG-NEUSTADT, NACHTWEIDE 36/43
VEB CARL ZEISS JENA 6900 JENA, CARL ZEISS-STRASSE 1
VEB MASCHINEN-UND STAHLBAU 8027 DRESDEN, HUFMUEHLENSTRASSE 5/15
VEB GUMMIWERK "JOHN SCHEHR" 3300 SCHOENEBECK, SCHILLERSTRASSE 5/7
VEB MASCHINEN-UND STAHLBAU 8027 DRESDEN, HUFMUEHLENSTRASSE 5/15
VEB MAGDEBURGER ARMATURENWERKE "KARL MARX" 3031 MAGDEBURG, LIEBKNECHTSTRASSE 65
VEB GUMMIWERK "JOHN SCHEHR" 3300 SCHOENEBECK, SCHILLERSTRASSE 5/7
VEB SCHWERMASCHINENBAU "KARL LIEBKNECHT" 3013 MAGDEBURG, ALT-SALBKE 6-10
VEB MASCHINENBAUHANDEL 3018 MAGDEBURG-NEUSTADT, NACHTWEIDE 36/43
VEB KOMBINAT PENTACON 8021 DRESDEN, SCHANDAUER STRASSE 76
VEB SCHWERMASCHINENBAU "GEORGI DIMITROFF 3011 MAGDEBURG, STR. D. DSF 82

```

```

DIE POSTLEITZAHLEN DER LIEFERANTEN:
1 3011 2 7010 3 3018 4 6900 5 8027 6 3300 7 8027 8 3031 9 3300 10 3013 11 3018
12 8021 13 3011

```

```

#####
POSTLEITZAHLEN AUS LIEFERANTEN-DATEI ERZEUGEN
#####

```

Anwendung der Strategie „LIFO“

DIE STRATEGIE "LIFO" (ZULETZT HINEIN - ZUERST HERAUS) IST ZUR UMKEHRUNG EINER ZEICHENKETTE ZU NUTZEN.

```

PROGRAM LIFO (INPUT,OUTPUT);
(* UMKEHRUNG EINER ZEICHENKETTE *)
(* DATENSATZTYP, ZEIGERVARIABLE *)
TYPE
  LISTE = @OBJEKT;
  OBJEKT = RECORD
    DATUM:CHAR;
    NACHFOLGE:LISTE
  END;
VAR
  I:INTEGER; ANFANG,P:LISTE;
BEGIN
  FOR I:=1 TO 32 DO WRITE('#'); WRITE(' ');
  BEGIN
    NEW(P);
    READ(P@,DATUM);
    WRITE(P@,DATUM);
    P@,NACHFOLGE:=ANFANG; (* "UMHAENGEN" DES *)
    ANFANG:=P (* ZEIGERS *)
  END; (* OF WHILE *)
  WKITELN;
  P:=ANFANG;
  WKITELN('AUSGABEDATEN: (LAST IN-FIRST OUT)');
  WHILE P <> NIL DO
    BEGIN
      WRITE(P@,DATUM);

```

```

WRITE('LAST IN - FIRST OUT '); P:=P@,NACHFOLGE
FOR I:=1 TO 32 DO WRITE('#'); WRITELN; END; (*OF WHILE *)
WRITELN('EINGABEDATEN: (FIRST IN=FIRST OUT)'); WRITELN;
ANFANG:=NIL; FOR I:=1 TO 103 DO WRITE('#'); WRITELN;
WHILE NOT EOF DO END. (*LIFU*)

```

```

##### LAST IN - FIRST OUT #####
EINGABEDATEN: (FIRST IN=FIRST OUT)
EIN NEGER MIT GAZELLE ZAGT IM REGEN NIE
AUSGABEDATEN: (LAST IN=FIRST OUT)
EIN NEGER MIT GAZELLE ZAGT IM REGEN NIE
#####

```

Dynamische Variable

DIE HANDHABUNG DYNAMISCHER VARIABLEN IST AN EINFACHEN BEISPIELEN ZU DEMONSTRIEREN. DIE WERTE DER ZEIGERVARIABLEN SIND IN HEXADEZIMALER DARSTELLUNG ANZUGEBEN.

```

PROGRAM ZEIGER(INPUT,OUTPUT);
(* DEMONSTRATION DYNAMISCHER VARIABLEN *)
(* ZEIGERTYP, ZEIGERVARIABLE *)
TYPE
BEZUGSTYP=INTEGER;
ZEIGERTYP=@BEZUGSTYP; (* "a"="PFEIL NACH OBEN" *)
VAR
I:INTEGER;
Z1,Z2,Z3,Z4:ZEIGERTYP; (* ZEIGERVARIABLE *)
BEGIN
FOR I:=1 TO 50 DO WRITE('='); WRITELN;
WRITELN('DEMONSTRATION DYNAMISCHER VARIABLEN:');
FOR I:=1 TO 50 DO WRITE('='); WRITELN;
WRITELN('HINWEISE:');
WRITELN('MIT DER DEKLARATION VON Z1,Z2,Z3 WERDEN WEDER VA-
RIABLE VOM TYP INTEGER VEREINBART NOCH SPEICHERPL.
RESERVIERT. SPEICHERPLATZ WIRD NUR DURCH "NEW" BE-
REITGESTELLT:');
WRITELN('HEXADEZIMALE DARSTELLUNG DER WERTE DER ZEIGERVARI-
ABLEN:');
Z1= 00094DFC
Z2= 00094DF8
Z3= 00094DF4
WRITELN('WERTZUWEISUNGEN AN DYNAMISCHE VARIABLE(BEZUGSVAR.)
ERGEBEN:');
BEZUGSVARIABLE Z1@= 1234
BEZUGSVARIABLE Z2@= 1234
BEZUGSVARIABLE Z1@= 12345678
BEZUGSVARIABLE Z2@= 87654321
BEZUGSVARIABLE Z3@= 10203040
Z1= 00094DF0
Z1= 00FFFFFF (=NIL)
FREIGABE VON SPEICHERPLATZ DURCH DIE PROZEDUR
"DISPOSE":
DISPOSE(Z3); DISPOSE(Z2);
NEW(Z4); WRITELN('NEW(Z4):',Z4:17,Z4:10 HEX);
FOR I:=1 TO 50 DO WRITE('='); WRITELN;
END. (* ZEIGER *)

```

Stücklistenzusammenstellung

FUER EIN ELEKTRISCHES GERÄT SIND DIE FUER DIE EINZELNEN BAUGRUPPEN ERFASSTEN BAUELEMENTE ZU EINER ZUSAMMENSTELLUNG NACH ARTIKELN ZU VERDICHTEN UND ZU DRUCKEN.

```

PROGRAM MATERIAL(INPUT,OUTPUT);
(* EINFACH VERKETTETE LISTE *)
(* ZEIGERTYP, ZEIGERVARIABLE *)
CONST
NL=13;
TYPE
BEZEICHNUNG=PACKED ARRAY[1..NL] OF CHAR;
LISTE=RECORD
NAME:BEZEICHNUNG;
NR:INTEGER;
BEGIN
NEW(H@,NACHFOLGER);
WITH H@,NACHFOLGER@ DO
BEGIN
NAME:=NAM;
NR:=NRO;
ANZAHL:=ANZ;
NACHFOLGER:=NIL;
END; (* WITH *)
END; (* ENDE D.N. EINTRAGUNG *)

```

```

ANZAHL:INTEGER;
NACHFOLGER:@LISTE
END; (* LISTE *)
LISTEZEI=@LISTE; (* POINTERTYP *)
VAR
I,J,NRO,ANZ:INTEGER; NAM:BEZEICHNUNG;
ANFANG:LISTEZEI; (* STATISCHE VARIABLE *)
PROCEDURE LIES;
VAR
I:INTEGER;
BEGIN
FOR I:=1 TO NL DO READ(NAM[I]);
READLN(NRO,ANZ);
END; (* LIES *)
PROCEDURE SUCH(NAM:BEZEICHNUNG;NRO,ANZ:INTEGER);
VAR
H:LISTEZEI;
BEGIN
IF ANFANG=NIL THEN (* LISTE IST LEER *)
BEGIN (* NEUE EINTRAGUNG *)
NEW(H);
H@,NAME:=NAM;
H@,NR:=NRO;
H@,ANZAHL:=ANZ;
H@,NACHFOLGER:=NIL;
ANFANG:=H; (* ZEIGER AUF LISTENANFANG *)
END ELSE (* LISTE IST NICHT LEER *)
BEGIN
H:=ANFANG; (* DURCHMUSTERUNG DER LISTE *)
WHILE (H@,NACHFOLGER<>NIL) AND (H@,NAME<>NAM)
DO H:=H@,NACHFOLGER;
IF H@,NAME=NAM THEN H@,ANZAHL:=H@,ANZAHL+ANZ
ELSE (* NEUE EINTRAGUNG *)
END; (* BEARBEITUNG D. N.LEEREN LISTE BEENDET *)
END; (* SUCH *)
PROCEDURE DRUCK; (* DRUCK DER LISTE *)
VAR
H:LISTEZEI; I:INTEGER;
BEGIN
I:=0; H:=ANFANG;
WHILE H<>NIL DO
BEGIN
I:=I+1; (* SPALTENZAehler *)
WRITE(H@,NAME:NL,H@,NR:9,H@,ANZAHL:3,' ');
H:=H@,NACHFOLGER; (* UEBERGANG ZUM N. ZEIGER *)
IF I MOD 4=0 THEN WRITELN; (* 4 SPALTEN *)
END
END; (* DRUCK *)
BEGIN
FOR I:=1 TO 103 DO WRITE('#'); WRITELN;
WRITELN('STUECKLISTE:62');
FOR I:=1 TO 103 DO WRITE('#'); WRITELN;
WRITELN('EINGABEDATEN:');
ANFANG:=NIL;
J:=0;
WHILE NOT EOF(INPUT) DO
BEGIN
J:=J+1; (* SPALTENZAehler ERHOEHEN *)
LIES;
SUCH(NAM,NRO,ANZ);
WRITE(NAM:NL,NRO:9,ANZ:3,' ');
IF J MOD 4=0 THEN WRITELN; (* 4 SPALTEN *)
END; (* WHILE *) WRITELN; WRITELN('=-:53');
WRITELN('ZUSAMMENSTELLUNG NACH ARTIKELN:67');
DRUCK;
END. (* MATERIAL *)

```

```

#####
STUECKLISTE
#####
EINGABEDATEN:
POTENTIOMETER 10001001 4 DIODE 40001001 10 WIDERSTAND 50001001 15 POTENTIUMETER 10001001 12
DIODE 40001001 15 TRANSISTOR 20001001 11 WIDERSTAND 50001001 30 DIODE 40001001 43
POTENTIUMETER 10001001 12 LEUCHTDIODE G 60001003 17 LEUCHTDIODE B 60001002 31 LEUCHTDIODE H 60001002 17
TRANSISTOR 20001001 21 TRANSISTOR 20001001 20 KONDENSATOR 30001001 25 KONDENSATOR 30001001 21
LEUCHTDIODE R 60001001 29 KONDENSATOR 30001001 2 LEUCHTDIODE G 60001003 17 WIDERSTAND 50001001 30
KONDENSATOR 30001001 2 LEUCHTDIODE B 60001002 12 WIDERSTAND 50001001 15
-.-
ZUSAMMENSTELLUNG NACH ARTIKELN:
POTENTIUMETER 10001001 28 DIODE 40001001 68 WIDERSTAND 50001001 90 TRANSISTOR 20001001 52
LEUCHTDIODE G 60001003 34 LEUCHTDIODE B 60001002 60 KONDENSATOR 30001001 50 LEUCHTDIODE R 60001001 29

```

Häufigkeitsanalyse von Vornamen

UM DEN SPEICHERPLATZ EINER UMFANGREICHEN DATEI EFFEKTIV ZU NUTZEN, SIND DIE VORNAMEN ALLER ERFASSTEN PERSONEN ZU VERSCHLUESSELN. ZUR VORBEREITUNG IST EINE ANALYSE DER HAEUFIGKEIT DES AUFTRETENS DIESER NAMEN VORZUNEHMEN.

```

PROGRAM ANALYSE(INPUT,OUTPUT);
(* HAEUFIGKEITSANALYSE VON NAMEN MIT BIN, BAUM *)
(* ZEIGERTYP, ZEIGERVARIABLE *)
TYPE
ZEIGT=@KNOTEN; (* ZEIGERTYP *)
KNOTEN=RECORD
NAME:ALFA;
ANZ:INTEGER;
LINKS,RECHTS:ZEIGT END; (* KNOTEN *)
VAR
I,J:INTEGER; NAM:ALFA; BAUM:ZEIGT;
PROCEDURE LIESALFA(VAR WORT:ALFA);
VAR
I:INTEGER;
BEGIN (* WERTZUWEISG. AN VARIABLE VOM TYP ALFA *)
FOR I:=1 TO 8 DO READ(WORT[I]);
END; (* LIESALFA *)
PROCEDURE SUCH(N:ALFA;VAR W:ZEIGT);
BEGIN (* DURCHSUCHEN, ANFUEGEN BZW. ERHOEHEN *)
IF W=NIL THEN
BEGIN (* EINRICHTEN EINES NEUEN KNOTENS *)
NEW(W);
WITH W@ DO BEGIN NAME:=N;
ANZ:=1;
END; (* ELSE *)
END; (* SUCH *)
PROCEDURE DRUCK(W:ZEIGT);
BEGIN (* DRUCK DES INHALTES ALLER KNOTEN *)
IF W<>NIL THEN BEGIN
DRUCK(W@,LINKS);
WRITE(W@,NAME,W@,ANZ:3,' ');
J:=J+1; (* DRUCKSPALTENZAehler *)
IF J MOD 8=0 THEN WRITELN;
DRUCK(W@,RECHTS);
END;
END; (* DRUCK *)
BEGIN
FOR I:=1 TO 51 DO WRITE('<>'); WRITELN('<');
WRITELN('HAEUFIGKEITSANALYSE VON VORNAMEN:67');
FOR I:=1 TO 51 DO WRITE('<>'); WRITELN('<');
WRITELN('EINGEGEBEN WURDEN FOLGENDE NAMEN:');
BAUM:=NIL;
I:=0; (* DRUCKSPALTEN=ZAEHLER *)
WHILE NOT EOF DO BEGIN
LIESALFA(NAM); READLN;
WRITE(NAM,' ');
I:=I+1;
IF I MOD 10=0 THEN WRITELN;

```

```

LINKS:=NIL;
RECHTS:=NIL;
END (* WERTZUWEISUNG AN KNOTEN *)
ELSE (* KNOTEN BEREITS VORHANDEN *)
BEGIN (* WERT-ERHOEHUNG DER KOMPONENTE ANZ *)
WITH w@ DO
IF N<NAME THEN SUCH(N,LINKS)
ELSE IF N>NAME THEN SUCH(N,RECHTS)
ELSE ANZ:=ANZ+1
END (* ELSE *)
END (* SUCH *)
SUCH(NAM,BAUM);
END; WRITELN;
WRITELN('=-:53);
WRITELN('ANALYSEN-ERGEBNISSE:':61);
J:=0; (* DRUCKSPALTEN-ZAEHLER *)
DRUCK(BAUM); WRITELN;
FOR I:=1 TO 51 DO WRITE('<>'); WRITELN('<');
END. (* ANALYSEN *)

```

```

HAEUFIGKEITSANALYSE VON VORNAMEN
EINGEGEBEN WURDEN FOLGENDE NAMEN:
MIKE HANNA BRIT HANS BRIT BRIT EDDA BRIT EDDA WOLFGANG
BRIT BRIT DORA BRIT BRIT HANNA MIKE BEN BEN
BEN BEN BEN BEN BEN ANNA ANNA HANNA DORA JENS
HANNA ANNI ANNI ANNI JENS UWE DORIS JENS JENS
HANNA GRIT HANNA HANNA EDDA EDDA THOMAS UWE
WOLFGANG WOLFGANG WOLFGANG KLAUS KLAUS DORA KLAUS KLAUS VOLKER VOLKER
VOLKER VOLKER WOLFGANG VOLKER MARION MARION MARITA MARITA ANETTE MARITA
BEN ANETTE GRIT GRIT GRIT GRIT THOMAS MIKE
MIKE THOMAS HANS BEN THOMAS ANDREAS WOLFGANG GRIT KURT ANDREAS
ANDREAS DORA ANDREAS ANDREAS ANETTE BRIT EDDA JENS MANFRED KURT
BERNT BEN BERNT GRIT GRIT GRIT WOLFGANG JENS GRIT GRIT GRIT
ANNA GRIT BEN JENS ANDREAS WOLFGANG HANNA GRIT GRIT GRIT
ANDREAS JENS DORIS MANFRED MANFRED MIKE MIKE UWE THOMAS THOMAS

```

ANALYSEN-ERGEBNISSE:

ANDREAS	7	ANETTE	3	ANNA	3	ANNI	5	BEN	12	BERNT	2	BRIT	9	DORA	4
DORIS	2	EDDA	6	GRIT	4	GRIT	12	HANNA	9	HANS	2	JENS	10	KLAUS	4
KURT	2	MANFRED	3	MARION	2	MARITA	3	MIKE	6	THOMAS	6	UWE	3	VOLKER	5
WOLFGANG	8														

Protokoll einer Häufigkeitsanalyse

IM PROGRAMM "ANALYSEN" IST DIE ARBEITSWEISE DER PROZEDUREN SUCH UND DRUCK DURCH DIE EINFUEGUNG VON ZUSÄTZLICHEN WRITE-ANWEISUNGEN (WERTE DER ZEIGERVARIABLEN, EINTRITTSKONTAKTE USW.) ZU "PROTOKOLLIEREN".

```

PROGRAM PROTANAL(INPUT,OUTPUT);
(* PROTOKOLL DER HAEUFIGKEITSANALYSE *)
(* ZEIGERTYP, ZEIGERVARIABLE *)
TYPE
ZEIGT=@KNOTEN; (* ZEIGERTYP *)
KNOTEN=RECORD
NAME:ALFA;
ANZ:INTEGER;
LINKS,RECHTS:ZEIGT END; (* KNOTEN *)
VAR
I:INTEGER; NAM:ALFA; BAUM:ZEIGT;
PROCEDURE LIESALFA(VAR WORT:ALFA);
VAR
I:INTEGER;
BEGIN (* WERTZUWEISG. AN VARIABLE VOM TYP ALFA *)
FOR I:=1 TO 8 DO READ(WORT[I]);
END; (* LIESALFA *)
PROCEDURE SUCH(N:ALFA;VAR W:ZEIGT);
BEGIN (* DURCHSUCHEN, ANFUEGEN BZW. ERHOEHEN *)
IF W=NIL THEN
BEGIN (* EINRICHTEN EINES NEUEN KNOTENS *)
NEW(W);
WRITE(W:8 HEX,' '); (* WERT D. ZEIGERVARIABLEN *)
I:=I+1; IF I MOD 10=0 THEN WRITELN;
WITH W@ DO BEGIN
NAME:=N;
ANZ:=1;
LINKS:=NIL;
RECHTS:=NIL;
END (* WERTZUWEISUNG AN KNOTEN *)
END ELSE (* KNOTEN BEREITS VORHANDEN *)
BEGIN (* WERT-ERHOEHUNG DER KOMPONENTE ANZ *)
WITH W@ DO
IF N<NAME THEN SUCH(N,LINKS)
ELSE IF N>NAME THEN SUCH(N,RECHTS)
ELSE ANZ:=ANZ+1
END (* ELSE *)
END; (* SUCH *)
PROCEDURE DRUCK(W:ZEIGT);
BEGIN (* DRUCK DES INHALTES ALLER KNOTEN *)
WRITE('>'); (* PROZEDUREINTRITT PROTOKOLLIERT. *)
IF W<NIL THEN BEGIN
WRITE(W@.LINKS:9 HEX,' L');
DRUCK(W@.LINKS);
WRITELN(' ':3,W@.NAME,W@.ANZ:4);
WRITE(W@.RECHTS:9 HEX,' R');
DRUCK(W@.RECHTS);
END;
WRITE('*'); (* PROZEDURAUSTRITT PROTOKOLLIERT. *)
END; (* DRUCK *)
BEGIN
FOR I:=1 TO 51 DO WRITE('<>'); WRITELN('<');
WRITELN('HAEUFIGKEITSANALYSE VON VORNAMEN:':67);
FOR I:=1 TO 51 DO WRITE('<>'); WRITELN('<');
WRITELN('EINGEGEBEN WURDEN FOLGENDE NAMEN:');
WRITE('HINWEIS:');
WRITE('DIE WERTE DER ZEIGERVARIABLEN WURDEN HIN-');
WRITE('ZUGEFUEGT BEI DER ERSTELLUNG DES BIN. ');
WRITE('BAUMES!'); WRITELN('');
BAUM:=NIL;
I:=0; (* DRUCKSPALTEN-ZAEHLER *)
WHILE NOT EOF DO BEGIN
LIESALFA(NAM); READLN;
WRITE(NAM,' ');
I:=I+1;
IF I MOD 10=0 THEN WRITELN;
SUCH(NAM,BAUM);
END; WRITELN;
WRITELN('=-:53);
WRITELN('ANALYSEN-ERGEBNISSE:':61);
WRITELN('BAUM = ',BAUM:9 HEX,' ');
DRUCK(BAUM); WRITELN;
FOR I:=1 TO 51 DO WRITE('<>'); WRITELN('<');
END. (* PROTANAL *)

```

```

HAEUFIGKEITSANALYSE VON VORNAMEN
EINGEGEBEN WURDEN FOLGENDE NAMEN:
(MITTEILUNG: DIE WERTE DER ZEIGERVARIABLEN WURDEN HINZUGEFUEGT BEI DER ERSTELLUNG DES BIN. BAUMES!)
MIKE 000A0DEC, HANNA 000A0DD8, BRIT 000A0DC4, HANS 000A0DB0, BRIT BRIT
EDDA 000A0D9C, BRIT EDDA WOLFGANG 000A0D88, BRIT DORA 000A0D74,
BRIT BRIT HANNA MIKE BEN 000A0D60, BEN BEN BEN
BEN BEN BEN ANNA 000A0D4C, ANNA HANNA DORA JENS 000A0D38,
HANNA ANNI 000A0D24, ANNI ANNI JENS UWE 000A0D10, DORIS 000A0CFC,
JENS JENS JENS HANNA GRIT 000A0CE8, HANNA HANNA HANNA EDDA
EDDA EDDA THOMAS 000A0CD4, UWE WOLFGANG WOLFGANG WOLFGANG KLAUS 000A0CC0,
KLAUS DORA KLAUS KLAUS VOLKER 000A0CAC, VOLKER VOLKER VOLKER WOLFGANG
VOLKER MARION 000A0C98, MARION MARITA 000A0C84, MARITA ANETTE 000A0C70, MARITA
BEN ANETTE GRIT 000A0C5C, GRIT GRIT JENS GRIT THOMAS
MIKE MIKE THOMAS HANS BEN THOMAS ANDREAS 000A0C48, WOLFGANG GRIT
KURT 000A0C34, ANDREAS ANDREAS DORA ANDREAS ANDREAS ANETTE BRIT FIDA
JENS MANFRED 000A0C20, KURT BERNT 000A0C0C, BEN BERNT GRIT GRIT
GRIT GRIT WOLFGANG JENS GRIT ANNA GRIT BEN JENS ANDREAS
WOLFGANG HANNA GRIT GRIT GRIT GRIT ANDREAS JENS DORIS MANFRED MANFRED
MIKE MIKE UWE THOMAS THOMAS

```

```

ANALYSEN-ERGEBNISSE:
BAUM = 000A0DEC
> 000A0DD8 L> 000A0DC4 L> 000A0D60 L> 000A0D4C L> 000A0C70 L> 000A0C48 L> 00FFFFFF L>* ANDREAS 7
00FFFFFF R>** ANETTE 3
00FFFFFF R>** ANNA 3
000A0D24 R> 00FFFFFF L>* ANNI 5
00FFFFFF R>*** BEN 12
000A0C0C R> 00FFFFFF L>* BERNT 2
00FFFFFF R>*** BRIT 9
000A0D9C R> 000A0D74 L> 00FFFFFF L>* DORA 4
000A0CFC R> 00FFFFFF L>* DORIS 2
00FFFFFF R>*** EDDA 6
000A0CE8 R> 000A0C5C L> 00FFFFFF L>* GRIT 4
00FFFFFF R>** GRIT 12
00FFFFFF R>*** HANNA 9
000A0DB0 R> 00FFFFFF L>* HANS 2
000A0D38 R> 00FFFFFF L>* JENS 10
000A0CC0 R> 00FFFFFF L>* KLAUS 4
000A0C98 R> 000A0C34 L> 00FFFFFF L>* KURT 2
000A0C20 R> 00FFFFFF L>* MANFRED 3
00FFFFFF R>*** MARION 2
000A0C84 R> 00FFFFFF L>* MARITA 5
00FFFFFF R>***** MIKE 6
000A0D88 R> 000A0D10 L> 000A0CD4 L> 00FFFFFF L>* THOMAS 6
00FFFFFF R>** UWE 3
000A0CAC R> 00FFFFFF L>* VOLKER 5
00FFFFFF R>*** WOLFGANG 8
00FFFFFF R>***

```

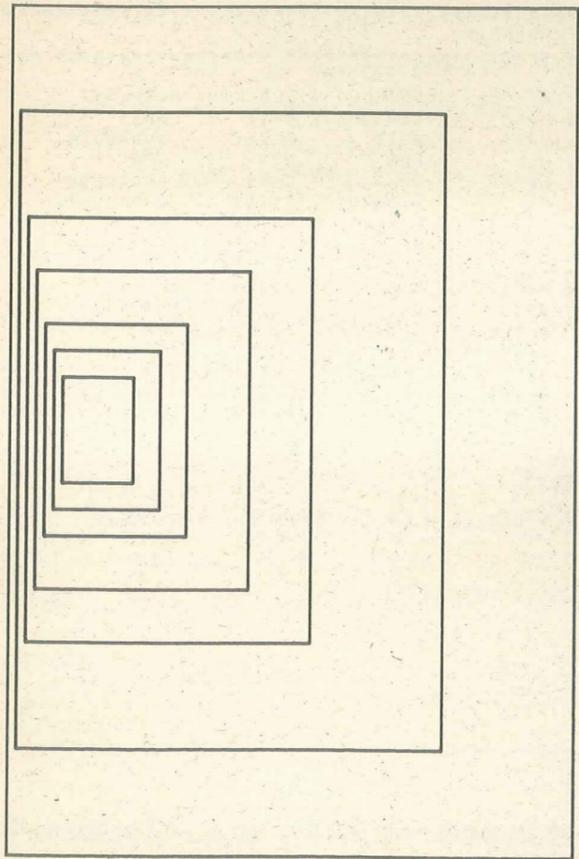
Optische Täuschungen

Abbildung siehe Seite 48
EINE AUSWAHL BEKANNTER EINFACHER OPTISCHER TÄUSCHUNGEN IST MITTELS DES PROGRAMMSYSTEMS PAD=ES GRAFISCH DARZUSTELLEN.

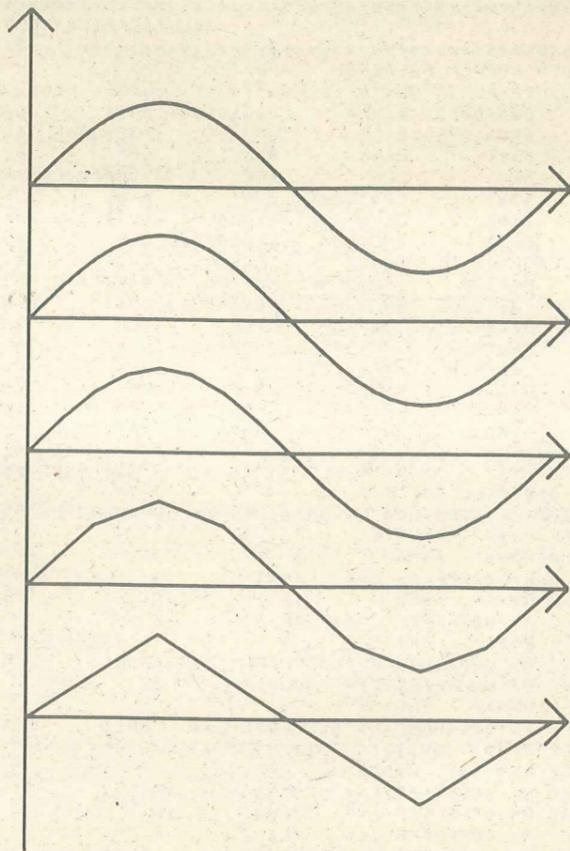
```

PROGRAM OPTISCH(OUTPUT);
(* OPTISCHE TAEUSCHUNGEN *)
(* PAD=ES, DIREKTIVEN *)
CONST
L=21;
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..L] OF CHAR;
VAR
I:INTEGER; X1,X2,Y:REAL;
PROCEDURE PBEGIN(T:ALF;XM,YM:REAL); EXTERNAL;
PROCEDURE ZLINE1(X1,Y1,X2,Y2:REAL); EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;T1:ALF; H,W:REAL); EXTERNAL;
PROCEDURE ZENDE; EXTERNAL;
BEGIN
PBEGIN('OPTI',130,200); (* ZEICHNUNGS-ANFANG *)
ZTEXT(0,0,19,'OPTISCHE TAEUSCHUNGEN',6,0,0,0);
ZLINE1(0,20,128,20); ZLINE1(128,20,128,70);
ZLINE1(128,70,0,70); ZLINE1(0,70,0,20);
ZLINE1(0,40,128,40); ZLINE1(128,40,0,40);
ZLINE1(0,50,128,50); ZLINE1(128,50,0,50);
FOR I:=1 TO 6 DO
BEGIN
X1:=10,0*(I-1);
X2:=128-X1;
ZLINE1(64,20,X1,45); ZLINE1(X1,45,64,70);
ZLINE1(64,70,X2,45); ZLINE1(X2,45,64,20);
END; (* FOR *)
FOR I:=1 TO 6 DO
BEGIN
Y:=80+20,0*(I-1);
IF ODD(I) THEN ZLINE1(14,Y,114,Y);
ELSE ZLINE1(114,Y,14,Y);
END; (* FOR *)
ZLINE1(0,194,14,180); ZLINE1(14,180,0,166);
ZLINE1(0,165,14,160); ZLINE1(14,160,0,155);
ZLINE1(28,154,14,140); ZLINE1(14,140,28,126);
ZLINE1(28,125,14,120); ZLINE1(14,120,28,115);
ZLINE1(0,110,14,100); ZLINE1(14,100,0,90);
ZLINE1(78,110,64,100); ZLINE1(64,100,78,90);
ZLINE1(100,110,114,100); ZLINE1(114,100,100,90);
ZLINE1(100,115,114,120); ZLINE1(114,120,100,125);
ZLINE1(100,126,114,140); ZLINE1(114,140,100,154);
ZLINE1(128,155,114,160); ZLINE1(114,160,128,165);
ZLINE1(128,166,114,180); ZLINE1(114,180,128,194);
ZENDE; (* ABSCHLUSS DER ZEICHENARBEIT *)
END. (* OPTISCH *)

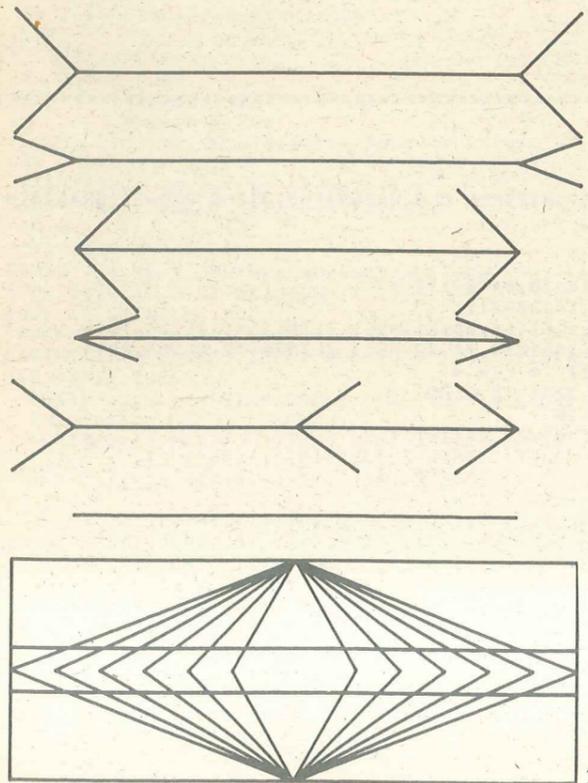
```



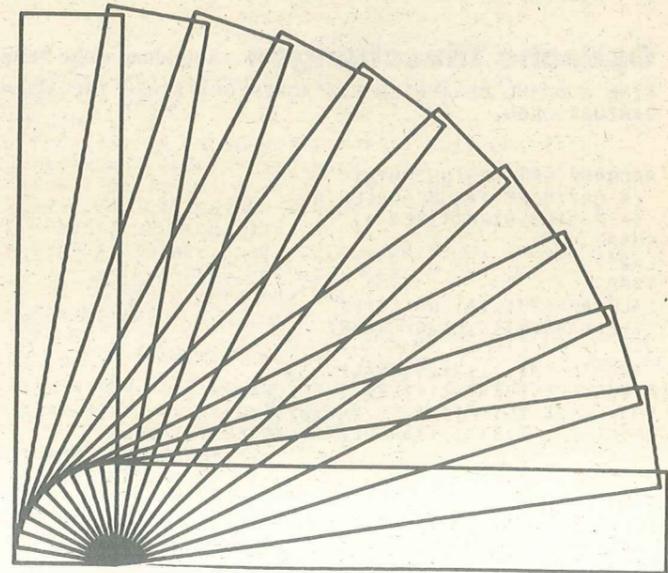
1



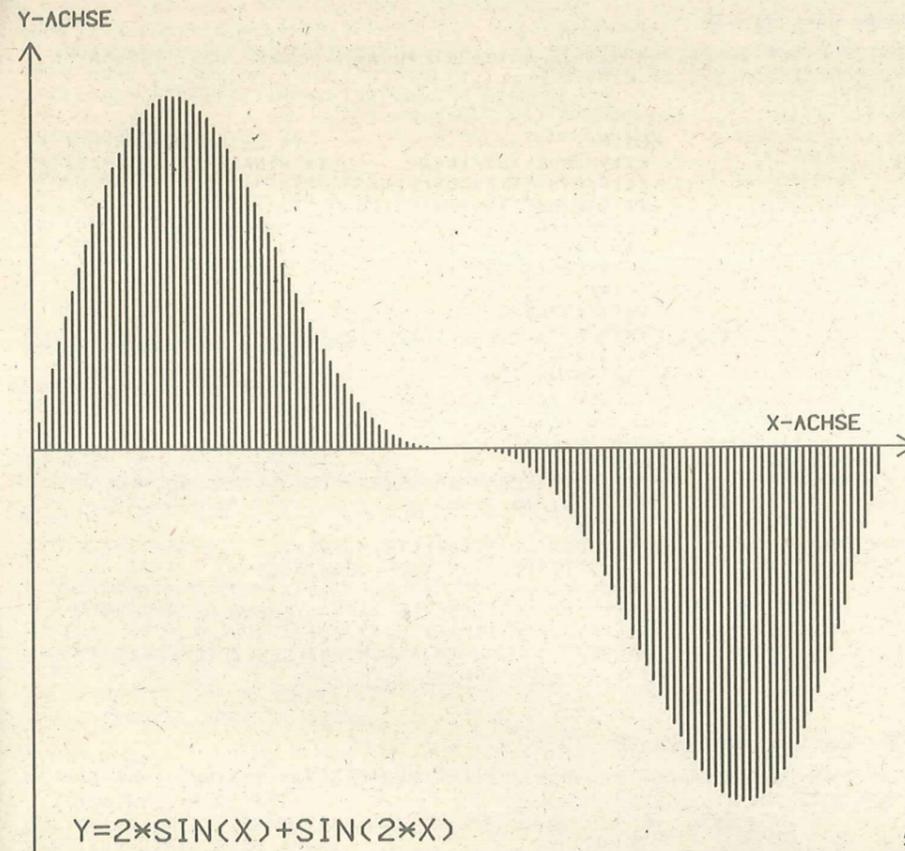
3



2



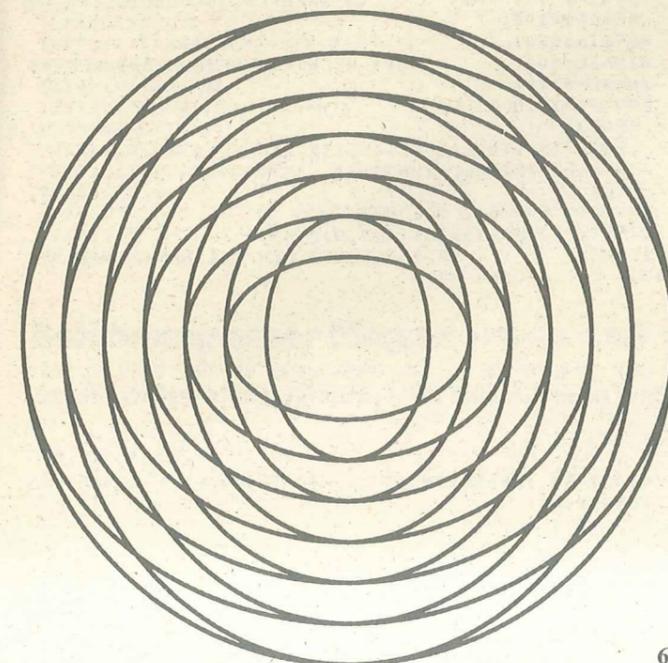
4



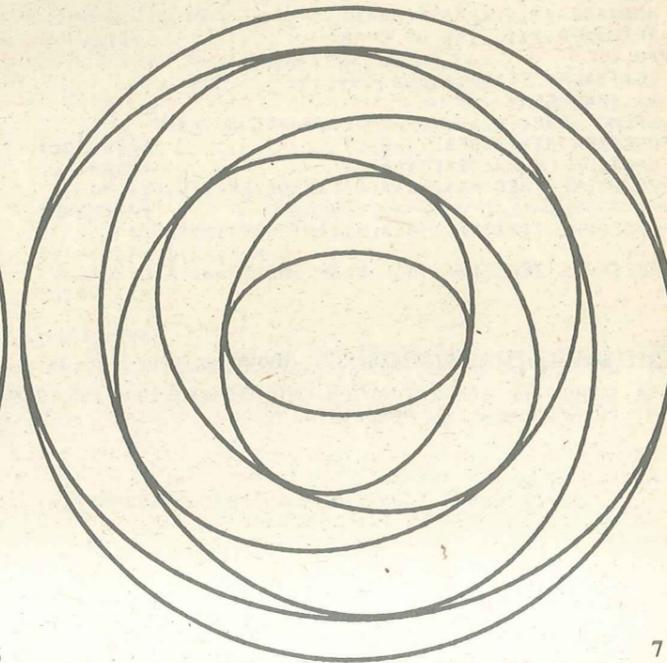
$$Y=2*\text{SIN}(X)+\text{SIN}(2*X)$$

5

- 1 Rechtecke im Raum (siehe Seite 50)
- 2 Optische Täuschung (siehe Seite 47)
- 3 Sinusapproximation (siehe Seite 50)
- 4 Erzeugung eines Fächers (siehe Seite 50)
- 5 Sinusfunktion (siehe Seite 51)
- 6 Verschachtelte Ellipsen (siehe Seite 53)
- 7 Gedrehte Ellipsen (siehe Seite 54)



6



7

Rechtecke im Raum Abbildung siehe Seite 48

IM RAUM STEHENDE RECHTECKE WERDEN UNTER EINEM SCHRAEGEN WINKEL BETRACHTET. MIT HILFE DES PROGRAMMSYSTEMS PAD-ES IST DAVON EINE ENTSPRECHENDE ZEICHNUNG ZU ERSTELLEN.

```
PROGRAM RECHTECK(INPUT,OUTPUT);
(* ZEICHNUNG: RECHTECK IM RAUM *)
(* PAD=ES, GRAFIK, PROZEDUR *)
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..36] OF CHAR;
VAR
TEXT:ALF; TEXTL:ALFL;
I,J:INTEGER;
X0,Y0,BREITE,HOEHE,WINKEL:REAL;
FUNCTION SIN(X:REAL):REAL;EXTERNAL;
FUNCTION COS(X:REAL):REAL;EXTERNAL;
(* VORAUSSETZUNG FUER DIE ARBEIT MIT PAD-ES 1 *)
PROCEDURE PBEGIN(TEXT:ALF;XM,YM:REAL); EXTERNAL;
PROCEDURE ZLINE1(X1,Y1,X2,Y2:REAL); EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;TEXTL:ALFL;
J,WINKEL:REAL); EXTERNAL;
PROCEDURE ZENDE; EXTERNAL;
PROCEDURE ZRECT(X0,Y0,BREITE,HOEHE,WINKEL:REAL);
(* RECHTECK MIT (X0,Y0) ALS LINKEM ECKPUNKT UND
"WINKEL" ALS NEIGUNGSWINKEL BEZUEGLICH X=ACHSE
IM PUNKT (X0,Y0) *)
CONST
PI=3,1415927;
VAR
FI,X1,Y1,XH,YH,X2,Y2,X3,Y3:REAL;
BEGIN
(* ZEICHNE RECHTECK *)
(* WINKEL IN GRADMASZ *)
FI:=WINKEL*PI/180;
X1:=X0+BREITE*COS(FI);
Y1:=Y0+BREITE*SIN(FI);
ZLINE1(X0,Y0,X1,Y1);
XH:=HOEHE*SIN(FI);
YH:=HOEHE*COS(FI);
X2:=X1-XH;
Y2:=Y1+YH;
ZLINE1(X1,Y1,X2,Y2);
X3:=X0-XH;
Y3:=Y0+YH;
ZLINE1(X2,Y2,X3,Y3);
ZLINE1(X3,Y3,X0,Y0);
END; (* ZRECT *)
BEGIN
PBEGIN('RECK',440,250); (* ZEICHNUNG=ANFANG *)
FOR I:=1 TO 7 DO
BEGIN
READLN(X0,Y0,BREITE,HOEHE);
ZRECT(X0,Y0,BREITE,HOEHE,0);
END;
TEXTL:='RECHTECKE IM RAUM STEHEND BETRACHTET';
ZTEXT(0,210,17,TEXTL,7,0);
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* RECHTECK *)
```

Erzeugung eines Fächers Abbildung siehe Seite 48

DURCH DREHUNG EINES RECHTECKS IST EIN FAECHER ZU ERZEUGEN UND MIT HILFE DES PROGRAMMSYSTEMS PAD-ES ZU ZEICHNEN.

```
PROGRAM FAECHER(OUTPUT);
(* DREHUNG EINES RECHTECKS UM EINEN PUNKT *)
(* PAD=ES, GRAFIK, EXTERNE PROZEDUR *)
CONST
XMAX=135; YMAX=135;
XM=21; YM=0;
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..13] OF CHAR;
VAR
TEXT:ALF; TEXT0:ALFL;
I,V:INTEGER;
LAENGE,BREITE,X0,Y0,WINKEL,ZUWACHS:REAL;
FUNCTION SIN(X:REAL):REAL; EXTERNAL;
FUNCTION COS(X:REAL):REAL; EXTERNAL;
PROCEDURE PBEGIN(TEXT:ALF;LAENGE,BREITE:REAL);
EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;T1:ALFL;
H,W:REAL); EXTERNAL;
PROCEDURE ZRECT(X0,Y0,LAENGE,BREITE,WINKEL:REAL);
EXTERNAL;
PROCEDURE ZENDE; EXTERNAL;
BEGIN
TEXT:='FAEC';
LAENGE:=XMAX; (* MAXIMALE ZEICHNUNG-LAENGE *)
BREITE:=YMAX; (* MAXIMALE ZEICHNUNG-BREITE *)
PBEGIN(TEXT,LAENGE,BREITE);
X0:=XM; Y0:=YM; (* DREHPUNKT-KOORDINATEN *)
LAENGE:=108;
BREITE:=20;
WINKEL:=0; (* RECHTECK-NEIGUNGSWINKEL *)
ZUWACHS:=9;
FOR I:=1 TO 11 DO
BEGIN
ZRECT(X0,Y0,LAENGE,BREITE,WINKEL);
WINKEL:=WINKEL+ZUWACHS;
END;
TEXT0:='F A E C H E R';
ZTEXT(25,120,13,TEXT0,8,0,0,0);
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* FAECHER *)
```

Sinusapproximation Abbildung siehe Seite 48

DIE SINUSFUNKTION IST DURCH UNTERSCHIEDLICHE POLYGONZUEGE NAEHERUNGSWEISE DARZUSTELLEN. DIESE ZEICHNUNG IST ENTSPRECHEND ZU BESCHRIFTEN.

```
PROGRAM SINUBILD(OUTPUT);
(* 5 NAEHERUNGSW. DARSTELLUNGEN DER SINUSFUNKT. *)
(* PAD=ES, EXTERNE PROZEDUREN *)
CONST
PI=3,1415926;
F=19;
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..13] OF CHAR;
VAR
I:ALF; T1:ALFL;
I,J,NENNER:INTEGER;
X1,Y1,X2,Y2,V,ZUWACHS,X:REAL;
FUNCTION SIN(X:REAL):REAL; EXTERNAL;
V:=0; (* VERSCHIEBUNG IN RICHTUNG DER Y-ACHSE *)
NENNER:=1;
FOR I:=1 TO 5 DO
BEGIN
NENNER:=2*NENNER;
ZUWACHS:=PI/NENNER;
X:=0;
X1:=5;
V:=V+30;
Y1:=V;
X2:=5;
ZLINE1(5,0,Y1,128,0,Y1); (* X=ACHSE *)
ZLINE1(128,0,Y1,123,0,Y1+5,0); (* SPITZE *)
ZLINE1(123,0,Y1+5,0,128,0,Y1);
END;
```

```
FUNCTION COS(X:REAL):REAL; EXTERNAL;
PROCEDURE PBEGIN(T:ALF;XM,YM:REAL); EXTERNAL;
PROCEDURE ZLINE1(X1,Y1,X2,Y2:REAL); EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;T1:ALFL;
H,W:REAL); EXTERNAL;
PROCEDURE ZENDE; EXTERNAL;
BEGIN
PBEGIN('SINU',128,0,300,0);
ZLINE1(5,0,0,0,5,0,190,0); (* Y=ACHSE *)
ZLINE1(5,0,190,0,10,0,185,0); (* SPITZE *)
ZLINE1(0,0,185,0,5,0,190,0);
ZTEXT(20,0,195,0,13,'SINUSFUNKTION',8,0,0,0);
FOR J:=1 TO 2*NENNER DO
BEGIN
(* NAEHERUNGSWEISE DARSTELLUNG *)
X:=X+ZUWACHS;
X2:=X2+F*ZUWACHS;
Y2:=V+F*SIN(X);
ZLINE1(X1,Y1,X2,Y2);
X1:=X2;
Y1:=Y2;
END; (* FOR J *)
END; (* FOR I *)
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* SINUBILD *)
```

Graphische Darstellung einer Sinusfunktion Abbildung siehe Seite 49

DIE FLAECHE UNTER EINER GEGEBENEN SINUSFUNKTION IST IN EINER GEEIGNETEN FORM GRAPHISCH DARZUSTELLEN UND ZU BESCHRIFTEN.

```
PROGRAM SINGRAF(OUTPUT);
(* ZEICHNUNG: DARSTELLUNG EINER SINUSFUNKTION *)
(* PAD=ES, GRAFIK *)
CONST
PI=3,14159265;
LAENGE=337; (* MAXIMALE LAENGE DER ZEICHNUNG *)
BREITE=263; (* MAXIMALE BREITE DER ZEICHNUNG *)
LANG=250;
F=40; (* DEHNUNGSFAKTOR *)
ABSTAND=2; (* ABSTAND DER STRICHE *)
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..20] OF CHAR;
VAR
TEXT:ALF;
TEXT0,TEXT1,TEXT2,TEXT3:ALFL;
I,V,PUNKTE:INTEGER;
X,Y,XA,Y0,XV,YV,XMAX,YMAX,DIST,WINKEL:REAL;
FUNCTION SIN(X:REAL):REAL; EXTERNAL;
FUNCTION COS(X:REAL):REAL; EXTERNAL;
PROCEDURE PBEGIN(T:ALF;XM,YM:REAL); EXTERNAL;
PROCEDURE ZLINE1(X1,Y1,X2,Y2:REAL); EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;T1:ALFL;
H,W:REAL); EXTERNAL;
PROCEDURE ZENDE; EXTERNAL;
BEGIN
TEXT:='SING';
TEXT0:='Y=2*SIN(X)+SIN(2*X)';
TEXT1:='X=ACHSE';
TEXT2:='Y=ACHSE';
TEXT3:='FUNKTIONS-GRAPHIK';
XMAX:=BREITE;
YMAX:=LAENGE;
PBEGIN(TEXT,XMAX,YMAX);
X:=0;
XV:=3; (* VERSCHIEBUNG IN X-RICHTUNG *)
YV:=130; (* VERSCHIEBUNG IN Y-RICHTUNG *)
Y0:=YV-120;
Y:=YV+120;
ZTEXT(15,15,19,TEXT0,6,0,0,0);
ZLINE1(XV,Y0,XV,Y); (* Y=ACHSE *)
Y0:=Y-5;
ZLINE1(X,Y0,XV,Y);
X:=6;
ZLINE1(XV,Y,X,Y0); (* SPITZE DER Y=ACHSE *)
ZTEXT(0,255,7,TEXT2,4,0,0,0);
ZTEXT(70,250,17,TEXT3,10,0,0,0);
X:=BREITE;
ZLINE1(XV,YV,X,YV); (* X=ACHSE *)
Y:=YV+3;
XA:=X-5;
ZLINE1(XA,Y,X,YV);
Y:=YV-3;
ZLINE1(X,YV,XA,Y); (* SPITZE DER X=ACHSE *)
ZTEXT(220,135,7,TEXT1,4,0,0,0);
Y0:=YV;
XA:=XV;
X:=0;
V:=1;
PUNKTE:=LANG DIV ABSTAND;
DIST:=2*PI/PUNKTE;
FOR I:=1 TO PUNKTE+1 DO
BEGIN
Y:=F*(2*SIN(X)+SIN(2*X))+YV;
IF ABS(Y-Y0) > 0,1 THEN
BEGIN
IF V=1 THEN ZLINE1(XA,Y0,XA,Y);
ELSE ZLINE1(XA,Y,XA,Y0);
V:=-V; (* RICHTUNGSWECHSEL *)
END; (* LAENGE EINER STRECKE MUSZ > 0 SEIN *)
X:=X+DIST;
XA:=XV+X*F;
END;
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* SINGRAF *)
```

Zeichnung einer Flagge Abbildung siehe IV. Umschlagseite

EINE FLAGGE IST ZU ENTWERFEN UND ZU ZEICHNEN, DIE AUS ZWEI KREISEN BESTEHT, EINE GEGEBENE ANZAHL VON PUNKTEN DES KREISES IST MIT DEN UEBRIGEN ZU VERBINDEN.

```
PROGRAM FLAGGE(OUTPUT);
(* ZEICHNUNG EINER FLAGGE MIT HILFE VON PAD-ES *)
(* PAD=ES, GRAFIK, PROZEDUR *)
CONST
NULL=0,0;
ANZAHL=32; (* ANZAHL DER PUNKTE AUF DEM KREIS *)
LAENGE=337; (* MAXIMALE LAENGE DER ZEICHNUNG *)
BREITE=263; (* MAXIMALE BREITE DER ZEICHNUNG *)
TYPE
ALF=ARRAY[1..4] OF CHAR;
VAR
TEXT:ALF;
PANZAHL:INTEGER; (* AKTUELLE PUNKT-ANZAHL *)
MX,MY,RADIUS,XMAX,YMAX,WINKEL,XU,YU:REAL;
X,Y:ARRAY[1..ANZAHL] OF REAL;
FUNCTION SIN(X:REAL):REAL; EXTERNAL;
END; (* DER ZU VERBINDENDEN PUNKTE DES KREISES *)
FOR I:=1 TO PUNKTE-1 DO
BEGIN
(* VERBINDE (X[I],Y[I]) MIT (X[J],Y[J]) *)
V:=1; (* STEUERT RICHTUNG DER VERBINDUNG *)
FOR J:=I+1 TO PUNKTE DO
BEGIN
IF V=1 THEN ZLINE1(X[I],Y[I],X[J],Y[J]);
ELSE ZLINE1(X[I],Y[I],X[J],Y[I]);
V:=-V; (* RICHTUNGSWECHSEL *)
END;
END;
END; (* VERBINDE *)
BEGIN
TEXT:='FLA1';
XMAX:=LAENGE;
YMAX:=BREITE;
```

```

FUNCTION COS(X:REAL):REAL;EXTERNAL;
(* VORAUSSETZUNG FUER DIE ARBEIT MIT PAD=ES ! *)
PROCEDURE PBEGIN(VAR A:ALF;VAR R,S:REAL);EXTERNAL;
PROCEDURE ZKREIS(VAR X,Y,R:REAL);EXTERNAL;
PROCEDURE ZLINE1(VAR X1,Y1,X2,Y2:REAL);EXTERNAL;
PROCEDURE ZRECT(VAR X1,Y1,XM,YM,W:REAL);EXTERNAL;
PROCEDURE ZENDE;EXTERNAL; (* ENDE DER ZEICHNUNG *)
PROCEDURE VERBINDE(KX,KY,RAD:REAL;PUNKTE:INTEGER);
(* VERBINDE EINE GEGEBENE ANZAHL VON PUNKTEN DES
KREISES MIT DEN MITTELPUNKTSKOORDINATEN KX,KY
UND DEM RADIUS RAD, ANZAHL ZU VERBINDENDER
PUNKTE GLEICH PUNKTE. *)
CONST
PI=3.14159265;
VAR
I,J,V:INTEGER;
BETA,FI:REAL;
BEGIN
BETA:=2*PI/PUNKTE; FI:=NULL;
FOR I:=1 TO PUNKTE DO
BEGIN
X[I]:=KX+RAD*COS(FI); (* BERECHNUNG *)
Y[I]:=KY+RAD*SIN(FI); (* DER KOORDINATEN *)
FI:=FI+BETA;
PBEGIN(TEXT,XMAX,YMAX);
XU:=NULL; (* DREHPUNKT DES RECHTECKES *)
YU:=NULL;
WINKEL:=NULL; (* DREHWINKEL DES RECHTECKES *)
ZRECT(XU,YU,XMAX,YMAX,WINKEL);
XU:=1;
YU:=1;
XMAX:=XMAX-4;
YMAX:=YMAX-4;
ZRECT(XU,YU,XMAX,YMAX,WINKEL);
MX:=LAENGE/2; (* MITTELPUNKT *)
MY:=BREITE/2; (* DER RECHTECKE *)
RADIUS:=BREITE/3;
ZKREIS(MX,MY,RADIUS); (* KREIS *)
PANZAHL:=ANZAHL;
VERBINDE(MX,MY,RADIUS,PANZAHL);
MX:=MX/3; (* KOORDINATEN *)
MY:=MY/3; (* DES KREISMITTELPUNKTES *)
RADIUS:=RADIUS/3;
ZKREIS(MX,MY,RADIUS); (* KLEINER KREIS *)
PANZAHL:=ANZAHL DIV 2;
VERBINDE(MX,MY,RADIUS,PANZAHL);
ZENDE; (* ABSCHLUSS DER ZEICHNUNG *)
END. (* FLAGGE *)

```

Zeichne Quadrate

Abbildung siehe Titelbild

DURCH DREHUNG EINES QUADRATES UM SEINEN MITTELPUNKT UND EINE SCHRITTWEISE VERKLEINERUNG DER SEITENLAENGE IST EINE FIGUR ZU ERZEUGEN.

```

PROGRAM QUADRATE(OUTPUT);
(* DREHUNG EINES QUADRATES *)
(* PAD=ES, GRAFIK, EXTERNE FUNKTION, PROZEDUR *)
CONST
L=20;
PI=3.14159265;
LAENGE=350;
BREITE=340;
PUNKTE=30;
S0=2;
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..L] OF CHAR;
VAR
TEXT:ALF;
I:INTEGER;
XMAX,YMAX,XM,YM,X,Y,W,FI,BETA,S,PV,R,K:REAL;
FUNCTION SORT(X:REAL):REAL;EXTERNAL;
FUNCTION SIN(X:REAL):REAL;EXTERNAL;
FUNCTION COS(X:REAL):REAL;EXTERNAL;
(* VORAUSSETZUNG FUER DIE ARBEIT MIT PAD=ES ! *)
PROCEDURE PBEGIN(T:ALF;XM,YM:REAL);EXTERNAL;
PROCEDURE ZRECT(X1,Y1,XM,YM,W:REAL);EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;TL:ALFL;
H,W:REAL);EXTERNAL;
PROCEDURE ZENDE;EXTERNAL;
BEGIN
TEXT:='QUAD';
K:=SQRT(2);
BETA:=PI/PUNKTE;
XMAX:=LAENGE;
YMAX:=BREITE;
YM:=YMAX/2-40;
XM:=XMAX/2-20;
PBEGIN(TEXT,XMAX,YMAX);
FI:=5*PI/4;
W:=0;
S:=5;
FOR I:=1 TO 2*PUNKTE DO
BEGIN
R:=S/K;
X:=XM-R*COS(FI);
Y:=YM-R*SIN(FI);
FI:=FI+BETA;
ZRECT(X,Y,S,S,W);
W:=I*3;
S:=S+S0;
END;
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* QUADRATE *)

```

Penrosesches Dreieck

Abbildung siehe III. Umschlagseite

DAS PENROSESCHES DREIECK IST ZU ERZEUGEN UND ZU ZEICHNEN. DABEI SOLL DER ZEICHENSTIFT "MOEGLICHS" WENIG ABGEHOEBEN WERDEN.

```

PROGRAM PENROSE(INPUT,OUTPUT);
(* PENROSESCHES DREIECK *)
(* PAD=ES, ZEIGERVARIABLE, RECORD *)
CONST
XMAX=140; YMAX=140;
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..20] OF CHAR;
LISTE=@PUNKT;
PUNKT=RECORD
LN:1..MAXINT;
X,Y:REAL;
PE:0..1; (* ENDE EINES KANTENZUGES=1 *)
NACHFOLGER:LISTE;
END;
VAR
TEXT:ALF; TEXTL:ALFL;
BEGIN
READLN(LN,X,Y,PE);
NACHFOLGER:=NIL;
END;
ELSE
BEGIN
WHILE P@.NACHFOLGER <> NIL DO
P:=P@.NACHFOLGER;
NEW(P@,NACHFOLGER);
WITH P@.NACHFOLGER@ DO
BEGIN
READLN(LN,X,Y,PE);
NACHFOLGER:=NIL;
END;
END;
END;

```

```

I:INTEGER;
LAENGE,BREITE,X1,Y1,X2,Y2:REAL;
ANFANG,P:LISTE;
PROCEDURE PBEGIN(TEXT:ALF;LAENGE,BREITE:REAL);
EXTERNAL;
PROCEDURE ZLINE1(X1,Y1,X2,Y2:REAL);
EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;N:INTEGER;TL:ALFL;
H,W:REAL);EXTERNAL;
PROCEDURE ZENDE;EXTERNAL;
BEGIN
TEXT:='PENR';
ANFANG:=NIL;
WHILE NOT EOF DO
BEGIN
IF ANFANG=NIL THEN
BEGIN
NEW(P);
ANFANG:=P;
WITH P@ DO
LAENGE:=XMAX;
BREITE:=YMAX;
PBEGIN(TEXT,LAENGE,BREITE);
P:=ANFANG;
WHILE P <> NIL DO
BEGIN
X1:=P@.X;
Y1:=P@.Y;
P:=P@.NACHFOLGER;
X2:=P@.X;
Y2:=P@.Y;
ZLINE1(X1,Y1,X2,Y2);
IF P@.PE<>0 THEN P:=P@.NACHFOLGER;
END;
TEXTL:='PENROSESCHES DREIECK';
ZTEXT(0,0,108,0,20,TEXTL,6,0,0,0);
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* PENROSE *)

```

Ein nicht möglicher Körper

Abbildung siehe III. Umschlagseite

ES IST EIN "NICHT MOEGLICHER KOERPER" ZU ERZEUGEN UND ZU ZEICHNEN. DIE ZEICHNUNG IST ENTSPRECHEND ZU BESCHRIFTEN.

```

PROGRAM KOERPER(INPUT,OUTPUT);
(* NICHT MOEGLICHER KOERPER *)
(* PAD=ES, DATENSATZ *)
CONST
N=46; (* ANZAHL ZU VERBINDENDER PUNKTE *)
M=40; (* ANZAHL DER STRECKEN *)
LAENGE=130; (* MAXIMALE LAENGE DER ZEICHNUNG *)
BREITE=220; (* MAXIMALE BREITE DER ZEICHNUNG *)
TYPE
PUNKT=RECORD X,Y:REAL END;
STRECKE=RECORD AP,EP:1..N END;
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..18] OF CHAR;
VAR
I:INTEGER;
TEXT:ALF; TEXTL:ALFL;
XMAX,YMAX,XU,YU,WINKEL,VX,VY:REAL;
P:ARRAY[1..N] OF PUNKT;
S:ARRAY[1..M] OF STRECKE;
FUNCTION SIN(X:REAL):REAL;EXTERNAL;
FUNCTION COS(X:REAL):REAL;EXTERNAL;
(* VORAUSSETZUNG FUER DIE ARBEIT MIT PAD=ES ! *)
PROCEDURE PBEGIN(TEXT:ALF;XM,YM:REAL);EXTERNAL;
PROCEDURE ZLINE1(X1,Y1,X2,Y2:REAL);EXTERNAL;
PROCEDURE ZRECT(X1,Y1,XM,YM,W:REAL);EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;I:INTEGER;TL:ALFL;
J,WINKEL:REAL);EXTERNAL;
PROCEDURE ZENDE;EXTERNAL;
BEGIN
READLN(VX); (* TRANSLATION IN RICHTUNG X-ACHSE *)
READLN(VY); (* TRANSLATION IN RICHTUNG Y-ACHSE *)
FOR I:=1 TO N DIV 2 DO
BEGIN
READ(P[2*I-1].X,P[2*I-1].Y,P[2*I].X,P[2*I].Y);
READLN; (* EINLESEN DER KOORDINATENWERTE *)
P[2*I-1].X:=P[2*I-1].X+VX;
P[2*I-1].Y:=P[2*I-1].Y+VY;
P[2*I].X:=P[2*I].X+VX;
P[2*I].Y:=P[2*I].Y+VY;
END; (* EINLESEN, ABSPEICHERUNG, TRANSLATION *)
(* DIE EINGABEDATEN WURDEN NICHT KORRIGIERT ! *)
FOR I:=1 TO M DO
READLN(S[I].AP,S[I].EP);
TEXT:='KOER'; XMAX:=LAENGE; YMAX:=BREITE;
PBEGIN(TEXT,XMAX,YMAX);
FOR I:=1 TO M DO
ZLINE1(P[S[I].AP].X,P[S[I].AP].Y,P[S[I].EP].X,
P[S[I].EP].Y);
TEXTL:='EIN NICHT';
XU:=100,0; YU:=40,0; WINKEL:=0,0;
ZTEXT(XU,YU,9,TEXTL,7,WINKEL);
YU:=30,0;
TEXTL:='MOEGLICHER KOERPER';
ZTEXT(XU,YU,18,TEXTL,7,WINKEL);
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* KOERPER *)

```

Verschachtelte Ellipsen

Abbildung siehe Seite 49

ELLIPSEN SIND SO INEINANDER ZU VERSCHACHTELN, DASS DIE GROSZE UND DIE KLEINE HALBACHSE "BENACHBARIET" ELLIPSEN ZUSAMMENFALLEN.

```

PROGRAM ELLIPSEN(OUTPUT);
(* VERSCHACHTELTE ELLIPSEN *)
(* PAD=ES, PROZEDUR *)
TYPE
ALF=ARRAY[1..4] OF CHAR;
ALFL=ARRAY[1..20] OF CHAR;
VAR
T:ALF; TL:ALFL;
I,J:INTEGER;
X,Y,A,B,W:REAL;
FUNCTION SIN(X:REAL):REAL;EXTERNAL;
FUNCTION COS(X:REAL):REAL;EXTERNAL;
(* VORAUSSETZUNG FUER DIE ARBEIT MIT PAD=ES ! *)
PROCEDURE PBEGIN(T:ALF;X,Y:REAL);EXTERNAL;
PROCEDURE ZLINE1(X,Y,X1,Y1:REAL);EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL;N:INTEGER;TL:ALFL;
H,W:REAL);EXTERNAL;
PROCEDURE ZENDE;EXTERNAL;
PROCEDURE ZELLIPSE(X,Y,A,B:REAL);
BEGIN
X1:=X+A*COS(W); (* KOORDINATEN DES 1. PUNKTES *)
Y1:=Y+B*SIN(W);
FOR I:=1 TO ANZAHL DO
BEGIN
W:=W+ZUWACHS; (* ARGUMENT D. TRIG.-FUNKTIONEN *)
X2:=X+A*COS(W); (* KOORDINATEN DES 2. PUNKTES *)
Y2:=Y+B*SIN(W);
ZLINE1(X1,Y1,X2,Y2); (* VERBINDUNGS-STRECKE *)
X1:=X2; (* KOORD. DES 1. PUNKTES DER JEWEILS *)
Y1:=Y2; (* FOLGENDEN STRECKE D. ELLIPSENBOGENS *)
END;
END; (* ZELLIPSE *)
BEGIN
PBEGIN('ELLI',150,150);
TL:='ELLIPSEN IN ELLIPSEN';
ZTEXT(0,0,20,TL,6,0);
W:=1.5708;
ZELLIPSE(64,80,64,64); (* EIN KREIS *)
A:=64; (* HALBACHSE *)

```

```
(* ZEICHNE ELLIPSE MIT MITTELPUNKT (X,Y), DEN
HALBACHSEN A UND B, DIE PARALELL ZUR X-,Y-ACHSE
SIND. *)
UNST
N:=2;
'AR
I, ANZAHL: INTEGER;
X1, Y1, X2, Y2, ZUWACHS: REAL;
EGIN
ANZAHL:=ROUND(N*(A+B)); (* STRECKENANZAHL=ELLIPSE *)
ZUWACHS:=6,283185/ANZAHL;
```

```
B:=56; (* HALBACHSE *)
FOR I:=1 TO 6 DO
  BEGIN
    ZELLIPSE(64,80,A,B);
    ZELLIPSE(64,80,B,A);
    A:=B;
    B:=B-8;
  END;
ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* ELLIPSEN *)
```

Gedrehte Ellipsen Abbildung siehe Seite 49

ELLIPSEN SIND UM IHREN SCHNITTPUNKT DER ACHSEN ZU DREHEN. BENACHBARTE ELLIPSEN SOLLTEN SICH BERUEHREN !

```
PROGRAM ELLIPSE(OUTPUT);
(* VERSCHACHTELTE GEDREHTE ELLIPSEN *)
(* PAD-ES, PROZEDUR *)
TYPE
  ALF=ARRAY[1..4] OF CHAR;
  ALFL=ARRAY[1..20] OF CHAR;
VAR
  T:ALF; TL:ALFL;
  I,J: INTEGER;
  X,Y,A,B,W: REAL;
FUNCTION SIN(X:REAL):REAL; EXTERNAL;
FUNCTION COS(X:REAL):REAL; EXTERNAL;
(* VORAUSSETZUNG FUER DIE ARBEIT MIT PAD-ES I *)
PROCEDURE PBEGIN(T:ALF; X,Y:REAL); EXTERNAL;
PROCEDURE ZLINE1(X,Y,X1,Y1:REAL); EXTERNAL;
PROCEDURE ZTEXT(X,Y:REAL; N:INTEGER; TL:ALFL;
  H,W:REAL); EXTERNAL;
PROCEDURE ZENDE; EXTERNAL;
PROCEDURE ZELLIPSE(X,Y,A,B,W:REAL);
(* ZEICHNE ELLIPSE MIT MITTELPUNKT (X,Y), DEN
HALBACHSEN A UND B UND DEM SCHNITTWINKEL W DER
HALBACHSE A MIT DER PARALLELEN ZUR X-ACHSE. *)
CONST
  N:=2;
VAR
  I, ANZAHL: INTEGER;
  X1, Y1, X2, Y2, FI, ZUWACHS: REAL;
BEGIN
  ANZAHL:=ROUND(N*(A+B)); (* TEILSTRECKEN *)
  FI:=0; (* ANFANGSWERT DES PARAMETERS *)
  ZUWACHS:=6,283185/ANZAHL;
```

```
X1:=X+A*COS(FI)*COS(W)-B*SIN(FI)*SIN(W);
Y1:=Y+A*COS(FI)*SIN(W)+B*SIN(FI)*COS(W);
(* KOORDINATEN DES 1. PUNKTES D, VERBINDUNGSSTR. *)
FOR I:=1 TO ANZAHL DO
  BEGIN
    FI:=FI+ZUWACHS;
    (* KOORDINATEN DES 2. PUNKTES DER VERB. STRECKE *)
    X2:=X+A*COS(FI)*COS(W)-B*SIN(FI)*SIN(W);
    Y2:=Y+A*COS(FI)*SIN(W)+B*SIN(FI)*COS(W);
    ZLINE1(X1,Y1,X2,Y2); (* VERBINDUNGS-STRECKE *)
    X1:=X2; (* KOORD. DES 1. PUNKTES DER JEWELNS *)
    Y1:=Y2; (* FOLGENDEN STRECKE D. ELLIPSENBOGENS *)
  END;
END; (* ZELLIPSE *)
BEGIN
  W:=0; (* HALBACHSEN PARALELL ZUR X,Y-ACHSE *)
  PBEGIN('ELLI',150,150);
  TL:='GEDREHTE ELLIPSEN';
  ZTEXT(0,0,17,TL,6,0);
  ZELLIPSE(64,80,64,64,0); (* EIN KREIS *)
  A:=64; (* HALBACHSE *)
  B:=56; (* HALBACHSE *)
  FOR I:=1 TO 6 DO
    BEGIN
      ZELLIPSE(64,80,A,B,W);
      W:=W+1,909859;
      A:=B;
      B:=B-8;
    END;
  ZENDE; (* ABSCHLUSS DER ZEICHEN-ARBEIT *)
END. (* ELLIPSE *)
```

Artikelliste

EINE FOLGE VON ARTIKELN, DIE DURCH EINE ARTIKELNUMMER EINDEUTIG IDENTIFIZIERT WERDEN, IST EINZULESEN UND ALS EINE LISTE ZU DRUCKEN, DEREN ELEMENTE NACH AUFSTIEGENDER ARTIKELNUMMER SORTIERT SIND.

```
PROGRAM LIST(INPUT,OUTPUT);
(* AUFBAU EINER ARTIKEL-LISTE DURCH EINFUEGEN *)
(* ZEIGERVARIABLE *)
TYPE
  ZEIGER=@LISTE;
  LISTE=RECORD
    NR: INTEGER; (* ARTIKEL-NUMMER *)
    NAME: ALFA; (* ARTIKEL-NAME *)
    NEXT: ZEIGER;
  END;
VAR
  J: INTEGER; STOP: BOOLEAN;
  ANFANG, ART, P: ZEIGER;
PROCEDURE STRICH;
VAR
  I: INTEGER;
BEGIN
  FOR I:=1 TO 103 DO WRITE('#'); WRITELN;
END; (* STRICH *)
PROCEDURE LIES(VAR N:ALFA);
VAR
  I: INTEGER;
BEGIN
  FOR I:=1 TO 8 DO READ(N[I]);
```

```
IF NOT EOF THEN BEGIN (* LISTE AUS 1 ARTIKEL *)
  NEW(ART);
  LIES(ART@,NAME);
  READ(ART@,NR);
  ART@,NEXT:=ANFANG;
  READLN;
  ANFANG:=ART;
END; (* NUR EIN ARTIKEL *)
WHILE NOT EOF DO
  BEGIN (* LIES UND FUEGE NEUEN ARTIKEL EIN *)
    NEW(ART);
    J:=J+1; (* ERHOEHUNG DES SPALTENZAEBLERS *)
    LIES(ART@,NAME); WRITE(ART@,NAME:11);
    READLN(ART@,NR); WRITE(ART@,NR:6);
    IF J MOD 6=0 THEN WRITELN;
    IF ANFANG@,NR>ART@,NR THEN
      BEGIN (* FUEGE ARTIKEL AM ANFANG EIN *)
        ART@,NEXT:=ANFANG;
        ANFANG:=ART;
      END
    ELSE
      BEGIN
        P:=ANFANG;
        STOP:=FALSE;
        WHILE (P@,NEXT<>NIL) AND (NOT STOP) DO
```

```
END; (* LIES *)
PROCEDURE DRUCK;
VAR
  J: INTEGER;
BEGIN
  (* AUSGABE DER LISTE *)
  (* SPALTENZAEBLER *)
  WHILE P<>NIL DO BEGIN
    WRITE(P@,NR:10,P@,NAME:10);
    P:=P@,NEXT;
    J:=J+1;
    IF J MOD 5=0 THEN WRITELN;
  END;
END; (* DRUCK *)
BEGIN
  STRICH;
  WRITELN('EINGABEDATEN:':58);
  J:=0; (* SPALTENZAEBLER *)
  ANFANG:=NIL;
```

```
IF P@,NEXT@,NR>ART@,NR THEN
  BEGIN (* FUEGE ARTIKEL IM INNERN EIN *)
    ART@,NEXT:=P@,NEXT;
    P@,NEXT:=ART;
    STOP:=TRUE;
  END
  ELSE P:=P@,NEXT;
  IF NOT STOP THEN BEGIN (* BZW. AM ENDE *)
    ART@,NEXT:=P@,NEXT;
    P@,NEXT:=ART;
  END (* ANFUEGUNG AM ENDE *)
END;
END; (* WHILE NOT EOF *)
WRITELN; (* DRUCK DER LETZTEN ZEILE *)
STRICH;
WRITELN('A R T I K E L - L I S T E.':65);
DRUCK; (* DRUCK DER ARTIKELLISTE *)
WRITELN;
STRICH;
END. (* LIST *)
```

```
#####
EINGABEDATEN:
ADAPTER 80011 DISKETTE 50101 MAUS 93032 LB-LESER 20006 LK-LESER 20009 TERMINAL 90010
DRUCKER 10042 PLOTTER 70008 MODEM 80004 STANZER 30055 GRIFFEL 90907
#####
ARTIKEL - LISTE:
10042 DRUCKER 20006 LB-LESER 20009 LK-LESER 30055 STANZER 50101 DISKETTE
60041 TABLET 70008 PLOTTER 80004 MODEM 80011 ADAPTER 90010 TERMINAL
90907 GRIFFEL 93032 MAUS
#####
```

Analyse von Pseudozufallszahlen

ES IST EIN "ZUFALLSZAHLEN-GENERATOR" ZU ENTWICKELN, DIE SO ERZEUGTEN PSEUDO-ZUFALLSZAHLEN SIND ZU ANALYSIEREN.

```
PROGRAM ZZAHLEN(OUTPUT);
(* HAEUFIGKEITSVERTEILUNG VON ZUFALLS-ZAHLEN *)
(* TRUNC, ROUND, FUNCTION, FOR=ANWEISUNG *)
CONST
  N=10000; (* ANZAHL DER AUSGEWERTETEN ZUFALLSZ. *)
  K=20; (* ANZAHL DER TEILINTERVALLE VON 0..1 *)
VAR
  I,J: INTEGER; X,Y,Z: REAL;
  KL: ARRAY[1..K] OF INTEGER; (* ZAEHLER *)
FUNCTION ZZ(Z0:INTEGER):REAL;
(* PSEUDO-ZUFALLSZAHLEN *)
  IF Z0 <> 0 THEN Z:=Z0/65536 ELSE Z:=253*Z;
  Z:=Z-TRUNC(Z); ZZ:=Z;
END; (* ZZ *)
BEGIN
  WRITELN('PSEUDO = Z U F A L L S Z A H L E N');
  Y:=ZZ(133); (* ERSTER AUFRUF DES ZZ-GENERATORS *)
  WRITELN('DER ERSTE AUFRUF LIEFERT DEN WERT: ',Y);
  FOR I:=1 TO 103 DO WRITE('-'); WRITELN;
  WRITE('GRAPHISCHE DARSTELLUNG DER HAEUFIGKEITS');
  WRITE('VERTEILUNG VON PSEUDOZUFALLSZAHLEN IM I');
  WRITELN('INTERVALL VON 0..1');
  FOR I:=1 TO 103 DO WRITE('-'); WRITELN;
  FOR I:=1 TO K DO KL[I]:=0;
  FOR I:=1 TO N DO
    BEGIN Y:=ZZ(0); (* WIEDERHOLTER GENERATORAUFR. *)
```

```
J:=TRUNC(K*Y)+1; (* INDEX D. TEILINTERV. *)
  KL[J]:=KL[J]+1 (* ERHOEHUNG D. ZAEHLERS *)
  END;
  WRITE(' :23); FOR I:=1 TO 6 DO WRITE(I:8,'00');
  WRITELN; (* BESCHRIFTUNG *)
  WRITE(' :22);
  FOR I:=1 TO 6 DO WRITE('-----+');
  WRITELN('-----> HAEUFIGKEIT'); (*ORDINATENACHSE*)
  FOR I:=1 TO K DO
    BEGIN WRITE(I:15,KL[I]:5,' ');
      FOR J:=1 TO ROUND(KL[I]/10) DO WRITE('*');
        WRITELN;
    END;
  WRITELN('TEIL- :15,' HAEUFIG-');
  WRITELN('INTERVALL:15,' KEIT ');
  FOR I:=1 TO 103 DO WRITE('-'); WRITELN;
  WRITE('BESTIMMUNG DER PERIODENLAENGE: ');
  I:=1; (* ZAEHLER *)
  X:=ZZ(0); Y:=ZZ(0);
  WHILE NOT (X=Y) DO
    BEGIN
      I:=I+1; Y:=ZZ(0);
    END;
  WRITE('DIE PERIODENLAENGE DER FOLGE DER ZZ IST');
  WRITELN(I:6,'. ');
  END. (* ZZAHLEN *)
```

```
PSEUDO = Z U F A L L S Z A H L E N
DER ERSTE AUFRUF LIEFERT DEN WERT: 2,02942E-03
-----
GRAPHISCHE DARSTELLUNG DER HAEUFIGKEITSVERTEILUNG VON PSEUDOZUFALLSZAHLEN IM INTERVALL VON 0..1:
-----
          100      200      300      400      500      600
-----> HAEUFIGKEIT
1 491 *****
2 523 *****
3 508 *****
4 494 *****
5 499 *****
6 494 *****
7 519 *****
8 522 *****
```



```

43 BOOLE LOGISCHE OPERATOREN * FOR-ANWEISUNG,BOOLEAN-TYP *
44 WHILEDEM ALPHABET * WHILE-ANWEISUNG,STANDARDFUNKTION *
45 FUNKTION TABELLIERUNG VERSCHIEDENER FUNKTIONEN * STANDARDFUNKTION,FUNKTION,FOR-ANWEISUNG *
46 WERBUNG WERBUNG FUER EIN STUDIUM * FELD,FOR-ANWEISUNG *
47 SOLLWERT MESZDATEN-AUSWERTUNG * FELDTYPEN,FOR-ANWEISUNG *
48 POTENZEN NEGATIVE POTENZEN VON 2 OHNE RUNDUNGSFEHLER * TYPDEFINITION,TEILBEREICHSTYPEN,FELD *
49 WERKZEUG LEBENSDAUER-HISTOGRAMM VON WERKZEUGEN * FELD,TEILBEREICHSTYPEN,FOR-ANWEISUNG *
50 SCHWINGE HARMONISCHE SCHWINGUNGEN M.VERSCH.DAEMPfung * FELD,FOR-ANWEISUNG,STANDARDFUNKTIONEN *
51 TAGE TAGE DER WOCHEN * CASE-ANWEISUNG,MENGETYP,MENGENOPERATOREN *
52 AEQUIVAL AEQUIVALENZ VON LOG. FUNKTIONEN * FOR-ANWEISUNG,BOOLESCHE OPERATOREN *
53 FELDER OPERATIONEN MIT EINDIM. FELDERN * FELDTYPEN,FOR-ANWEISUNG *
54 VIERPOL VIERPOLMATRIZEN * FELDTYPEN,FOR-ANWEISUNG *
55 MATRIX ZWEIDIMENSIONALES FELD AUSGEBEN * FELD,FOR-ANWEISUNG,VERBUNDANWEISUNG,WRITELN *
56 KREISTAB TABELLE VON GROESZEN DES KREISES * FUNKTIONSDOKUMENTATION,FUNKTIONSAUFRUF *
57 ELEFUNKT ZUORDNUNGEN VON KONSTANTEN * FUNKTION,CASE-ANWEISUNG,FOR-ANWEISUNG *
58 EIGENBAU FUNKTIONEN VON VERSCHIEDENEM TYP * FUNKTIONSDOKUMENTATION *
59 HAUS SKIZZE EINES HAUSES * FOR-ANWEISUNG,IF-ANWEISUNG,*
60 MARKE UNBEDINGTER SPRUNG * MARKE,GOTO-ANWEISUNG *
61 DREIECKE DREIECKSFLAECHE * FUNKTION,FUNKTIONSAUFRUF,LOKAL *
62 FAKULTAE REKURSIVE U.ITERATIVE FAKULTAETSBERECHNUNG * FUNKTION,LOKAL *
63 STANDARD STANDARD-PROZEDUREN/FUNKTIONEN=UEBERSICHT * PROZEDUR *
64 ALFAWORT EIN-U. AUSGABE VON WERTEN VON ALFA-GROESZEN * ALFATYP,READLN,PROZEDUR,LOKAL *
65 VEKTOREN VEKTOR-RECHNUNG * FUNKTION,PROZEDUR,FOR-ANWEISUNG *
66 REKURSION SELBST-REKURSION * PROCEDURE,LOKAL *
67 GAESTE SITZORDNUNGEN VON N GAESTEN * PROZEDUR *
68 GEGENSEITIGE REKURSION * PROZEDUR,PROZEDURAUFRUF,DIREKTIVE *
69 CARDCOPY PROTOKOLLIERUNG VON LOCHKARTEN-INHALTEN * TEXTFILE,WHILE-ANWEISUNG,READLN,WRITELN *
70 DATENTYP UEBERSICHT UEBER DATENTYPEN * PROZEDUR,TEXTFILE *
71 MITTEL ITERATIVE MITTELWERTBERECHNUNG * PROCEDURE *
72 KOMPLEX ARITHMETIK KOMPLEXER ZAHLEN * ARRAY, RECORD, BOOLEAN, PROCEDURE *
73 HOEKER ARBEIT MIT EINER TEILNEHMER-LISTE * DATENSATZTYPEN,FILE,PROZEDUR *
74 KARTEI PERSONEN-DATEI * DATENSATZTYPEN,WITH-ANWEISUNG *
75 VOLUMINA VOLUMEN VON KUGEL,ZYLINDER * DATENSATZTYPEN,PROZEDUR,MARKEN *
76 KUENSTE VARIABLE SAETZE VON KUENSTLERDATEN * DATENSATZTYPEN,CASE-ANWEISUNG,VARIANTENTEIL *
77 PLZAHL ERZEUGUNG VON POSTLEITZAHLEN AUS TEXTFILE * FILETYPEN,PUFFERVARIABLE,GET,PUT *
78 LIFO UMKEHRUNG EINER ZEICHENKETTE * DATENSATZTYP,ZEIGERVARIABLE *
79 ZEIGER DEMONSTRATION DYNAMISCHER VARIABLEN * ZEIGERTYP,ZEIGERVARIABLE *
80 MATERIAL EINFACH VERKETTETE LISTE * ZEIGERTYP,ZEIGERVARIABLE *
81 ANALYSEN HAEUFIGKEITSANALYSE VON NAMEN MIT BIN. BAUM * ZEIGERTYP,ZEIGERVARIABLE *
82 PROTANAL PROTOKOLL DER HAEUFIGKEITSANALYSE * ZEIGERTYP,ZEIGERVARIABLE *
83 OPTISCH OPTISCHE TAEUSCHUNGEN * PAD-ES,DIREKTIVEN *
84 RECHTECK ZEICHNUNG: RECHTECKE IM RAUM * PAD-ES,GRAFIK,PROZEDUR *
85 FAECHER DREHUNG EINES RECHTECKS UM EINEN PUNKT * PAD-ES,GRAFIK,EXTERNE PROZEDUR *
86 SINUBILD 5 NAEHERUNGSW. DARSTELLUNGEN DER SINUSFUNKT.* PAD-ES,EXTERNE PROZEDUREN *
87 SINGRAF ZEICHNUNG: DARSTELLUNG EINER SINUSFUNKTION * PAD-ES,GRAFIK *
88 FLAGGE ZEICHNUNG EINER FLAGGE MIT HILFE VON PAD-ES * PAD-ES,GRAFIK,PROZEDUR *
89 QUADRATE DREHUNG EINES QUADRATES * PAD-ES,GRAFIK,EXTERNE FUNKTION,PROZEDUR *
90 PENROSE PENROSESCHES DREIECK * PAD-ES, ZEIGERVARIABLE, RECORD *
91 KOERPER NICHT MOEGLICHER KOERPER * PAD-ES,DATENSATZ *
92 ELLIPSEN VERSCHACHTELTE ELLIPSEN * PAD-ES,PROZEDUR *
93 ELLIPSE VERSCHACHTELTE GEDREHTE ELLIPSEN * PAD-ES,PROZEDUR *
94 LIST AUFBAU EINER ARTIKEL-LISTE DURCH EINFUEGEN * ZEIGERVARIABLE *
95 ZZAHLEN HAEUFIGKEITSVERTEILUNG VON ZUFALLS-ZAHLEN * TRUNC, ROUND, FUNKTION, FOR-ANWEISUNG *
96 INHALT VERZEICHNIS DER PROGRAMME MIT INHALTSANGABE * PROZEDUR,PAGE,DATE *
97 REGISTER VERZEICHNIS DER PROGRAMME UND AUFGABEN * ZEIGERVARIABLE,PAGE,DATE *
98 EFFEKT SEITENEFFEKT BEIM FUNKTIONSAUFRUF * FUNKTION,FUNKTIONSAUFRUF *
99 EFFECT SEITENEFFEKT BEI DER MULTIPLIKATION * FUNKTION *
100 SACHWORT SACHWORTVERZEICHNIS ZU PASCAL-PROGRAMMEN * HINAERER BAUM,LISTE *
101 FURLOOP AENDERUNG DES ENDWERTES EINER LAUFVARIABLEN * FOR-ANWEISUNG,AUFZAEHLUNGSTYP,ENDWERT *
#####
08.09.85

```

Verzeichnis der Programme und Aufgabenstellungen

ES IST EIN ALPHABETISCHES REGISTER DER PROGRAMM-NAMEN UND DER ZUGEORDNETEN AUFGABENSTELLUNGEN ZU ERZEUGEN UND IN ZWEI SPALTEN ZU DRUCKEN.

```

PROGRAM REGISTER(INPUT,OUTPUT);
(* VERZEICHNIS DER PROGRAMME UND AUFGABEN *)
(* ZEIGERVARIABLE, PAGE, DATE *)
CONST
  LG=48;
TYPE
  ALFL=ARRAY[1..LG] OF CHAR;
  ZEIGER=@LIST;
  LIST=RECORD
    NR:1..250; (* LAUFENDE NR. DES PROGRAMMS *)
    PN:ALFA; (* PROGRAMMNAMEN *)
    AB:ALFL; (* AUFGABEN-BEZEICHNUNG *)
    NEXT:ZEIGER
  END;
  ELSE WRITELN(' '); I:=I+1;
  WRITE(' ':50-LG); DRUCKALF(P@,AB);
  WRITE(' ':53-LG);
  IF PH<>NIL THEN DRUCKALF(PH@,AB); WRITELN;
  IF I=30 THEN PAGE(OUTPUT);
  IF I MOD 32=30 THEN PAGE(OUTPUT);
  P:=P@,NEXT; (* ZEIGER AUF LINKE DRUCKSPALTE *)
  IF PH<>NIL THEN PH:=PH@,NEXT; (* R. SPALTE *)
END; (* WHILE *)
END; (* DRUCK *)
BEGIN
  PAGE(OUTPUT); STRICH;
  WRITELN('PROGRAMM- UND AUFGABENREGISTER':66);

```

```

END;
VAR
  I,J,K,L:INTEGER; STOP:BOOLEAN; D:ALFA;
  ANF,PROG,P,PH:ZEIGER;
PROCEDURE STRICH;
VAR
  I:INTEGER;
BEGIN
  (* DRUCKE STRICH *)
  FOR I:=1 TO 103 DO WRITE(' '); WRITELN;
END; (* STRICH *)
PROCEDURE LIESALFA(VAR A:ALFA);
VAR
  I:INTEGER;
BEGIN
  (* LIES WERT DER VARIABLEN VOM TYP ALFA *)
  FOR I:=1 TO 8 DO READ(A[I]);
END; (* LIESALFA *)
PROCEDURE LIESALFL(VAR A:ALFL);
VAR
  I:INTEGER;
BEGIN
  (* LIES WERT DER VARIABLEN VOM TYP ALFL *)
  FOR I:=1 TO LG DO READ(A[I]);
END; (* LIESALFL *)
PROCEDURE DRUCKALF(B:ALFL);
VAR
  I:INTEGER;
BEGIN
  FOR I:=1 TO LG DO WRITE(B[I]);
END; (* DRUCKALF *)
PROCEDURE MITTE;
BEGIN
  (* BESTIMME "MITTE" DER LISTE *)
  I:=1;
  P:=ANF; PH:=ANF;
  WHILE P<>NIL DO BEGIN
    P:=P@,NEXT;
    IF I=1 THEN PH:=PH@,NEXT; I:=I+1;
  END; PROG:=PH; (* "MITTE" *)
END; (* MITTE *)
PROCEDURE DRUCK;
BEGIN
  (* ZWEISPALTIGER REGISTER-DRUCK *)
  P:=ANF; I:=0; (* ANFANG DER LISTE BZW ZAEHLERS *)
  WHILE P<>PROG DO
    BEGIN
      WRITE(P@,PN:8,P@,NR:42,' ':3); (* L. SPALTE *)
      IF PH<>NIL THEN WRITELN(PH@,PN:8,PH@,NR:42)
    END;

```

```

STRICH;
I:=1; (* PROGRAMM-ZAEHLER *)
ANF:=NIL;
IF NOT EOF THEN BEGIN
  (* LISTE AUS 1 PROGR. *)
  NEW(PROG);
  PROG@,NR:=I; I:=I+1;
  LIESALFA(PROG@,PN); READLN;
  LIESALFL(PROG@,AB); READLN;
  PROG@,NEXT:=ANF;
  ANF:=PROG;
END;
WHILE NOT EOF DO
  BEGIN
    (* LIES UND FUEGE NEUES PROGRAMM EIN *)
    NEW(PROG);
    PROG@,NR:=I; I:=I+1;
    LIESALFA(PROG@,PN); READLN;
    LIESALFL(PROG@,AB); READLN;
    IF ANF@,PN>PROG@,PN THEN
      BEGIN
        (* FUEGE NEUES PROGRAMM AM ANFANG EIN *)
        PROG@,NEXT:=ANF;
        ANF:=PROG;
      END
    ELSE
      BEGIN
        P:=ANF;
        STOP:=FALSE;
        WHILE (P@,NEXT<>NIL) AND (NOT STOP) DO
          IF P@,NEXT@,PN>PROG@,PN THEN
            BEGIN
              (* FUEGE PROGRAMM IM INNERN EIN *)
              PROG@,NEXT:=P@,NEXT;
              P@,NEXT:=PROG;
              STOP:=TRUE;
            END
          ELSE P:=P@,NEXT;
        IF NOT STOP THEN BEGIN
          PROG@,NEXT:=P@,NEXT;
          P@,NEXT:=PROG;
        END (* ANFUEGUNG AM ENDE *)
      END
    END (* IF ANF@,PN>... *)
  END; (* WHILE NOT EOF *)
J:=1-1; (* ANZAHL DER PROGRAMME *)
MITTE; DRUCK; WRITELN; STRICH;
WRITE('ANZAHL DER PROGRAMME':J,4);
DATE(D); WRITELN('DATUM':68,U:10); (* DATUM *)
END. (* REGISTER *)

```

PROGRAMM- UND AUFGABENREGISTER

ABSTAND	25	MASSE	36
ABSTAND ZWEIER PUNKTE		MASSE-BERECHNUNG EINES ROHTEILS	
AEQUIVAL	52	MATERIAL	79
ZWEI SCHALTUNGEN		STUECKLISTENZUSAMMENSTELLUNG	
ALFATYP	30	MATRIX	55
MANIPULATION MIT WORTEN		DARSTELLUNG EINER MATRIX	
ALFAWORT	64	MITTEL	71
PASCAL-STANDARDBEZEICHNER		ITERATIVE BERECHNUNG DES ARITHMETISCHEN MITTELS	
ANALYSEN	81	NAMEN	34
HAEUFIGKEITSANALYSE VON VORNAMEN		SIGNIFIKANTE STELLEN	
AUSGABE	15	NULLTEIL	35
BUCHSE		NULLTEILER	
BILDAUS	41	OPERATIO	9
BILDAUSWERTUNG		RECHENOPERATIONEN MIT KONSTANTEN	
BOOLE	43	OPTISCH	83
BOOLEISCHE GROESZEN		OPTISCHE TAEUSCHUNGEN	
CARDCOPY	69	PENROSE	90
KOPIE DES INHALTES VON LOCHKARTEN		PENROSESCHES DREIECK	
DATENTYP	70	PLANIST	16
HIERARCHIE DER DATENTYPEN		PLAN-ABRECHNUNG	
DREIECKE	61	PLZAHL	77
DREIECKSFLAECHE		POSTLEITZAHLEN	
EFFECT	99	POTENZEN	48
SEITENEFFEKT BEI DER MULTIPLIKATION		ZWEIERPOTENZEN	
EFFECT	98	PRODUKTE	26
DEMONSTRATION EINES SEITENEFFEKTES		SUMME VON PRODUKTEN	
EIGENBAU	58	PROGRAMM	7
RATIONALISIERUNG DER PROGRAMMIERUNG		SYNTAX EINES PROGRAMMS	
ELEFUNKT	57	PROTANAL	82
INVERSE ELEMENTARE FUNKTIONEN		PROTOKOLL EINER HAEUFIGKEITSANALYSE	
ELLIPSE	93	PROVERB	6

33 EIN-/AUSGABE-PROZEDUREN 34 EINFACHE ANWEISUNG 35 EINFACHER TYP 36 EINGABE
 37 ENDWERT 38 ENTPACKEN 39 ERGIBTANWEISUNG 40 FALLKONSTANTENLISTE
 41 FELD 42 FELDBREITE 43 FELDTYPEN 44 FESTER TEIL
 45 FILE 46 FILEPUFFER 47 FILETYPEN 48 FOR-ANWEISUNG
 49 FORMALE PARAMETER 50 FORTRAN 51 FORWARD-DIREKTIVE 52 FUNKTION
 53 FUNKTIONSAUFRUF 54 FUNKTIONSDOKUMENTATION 55 GEPACKTE DATENTYPEN 56 GET
 57 GLOBAL 58 GOTO-ANWEISUNG 59 GRAFIK 60 GUELTIGKEITSBEREICH
 61 IF-ANWEISUNG 62 INDEXTYP 63 INDIZIERTE VARIABLE 64 INTEGER-TYP
 65 KOMMENTAR 66 KOMPONENTENTYP 67 KOMPONENTENVARIABLE 68 KONSTANTE
 69 KONSTANTENDEFINITION 70 KONSTANTENDEFINITIONSTEIL 71 KONVERTIERUNG 72 LAUFVARIABLE
 73 LEERE ANWEISUNG 74 LISTE 75 LOKAL 76 MARKEN
 77 MARKENDEKLARATIONSTEIL 78 MENGE 79 MENGENOPERATOREN 80 MENGENTYP
 81 MORPHEM 82 NAME 83 ORDINALTYP 84 OPERATOREN
 85 PACKEN 86 PAD-ES 87 PAGE 88 PARAMETER
 89 PARAMETERLISTE 90 PARAMETERUEBERGABE 91 PARAMETERKVERTRAEGELICHKEIT 92 PROGRAMM
 93 PROGRAMMAUFBAU 94 PROGRAMMKOPF 95 PROGRAMMPARAMETER 96 PROZEDUR
 97 PROZEDURAUFRUF 98 PROZEDURKOPF 99 PUFFERVARIABLE 100 PUT
 101 QUELLTEXT 102 READ 103 READLN 104 REAL-TYP
 105 REKURSIV 106 REPEAT-ANWEISUNG 107 RESET 108 RESULTATTYP
 109 ROUTINE 110 ROUTINEAUFRUF 111 ROUTINEDEKLARATIONSTEIL 112 ROUTINEKOEPE
 113 ROUTINEPARAMETER 114 REWRITE 115 SCHACHTELUNG 116 SIMULATION
 117 SORTIERUNG 118 SPEZIALSYMBOL 119 STANDARDBEZEICHNER 120 STANDARDFILE
 121 STANDARDFUNKTIONEN 122 STANDARDTYPEN 123 STRING 124 STRUKTURIERTE ANWEISUNG
 125 STRUKTURIERTE TYPEN 126 SYNTAX 127 TEILBEREICHSTYPEN 128 TERM
 129 TEXTFILE 130 TRANSKRIPTION 131 TRENNZEICHEN 132 TYP
 133 TYPDEFINITION 134 TYPDEFINITIONSTEIL 135 VARIABLE 136 VARIABLENDEKLARATION
 137 VARIABLENDEKLARATIONSTEIL 138 VARIABLENPARAMETER 139 VARIANTENTEIL 140 VERBUNDANWEISUNG
 141 VEREINBARUNGSTEIL 142 VERGLEICHOPERATOREN 143 VOLLSTAENDIGE VARIABLE 144 VORZEICHENL. GANZE ZAHL
 145 VORZEICHENLOSE KONSTANTE 146 VORZEICHENLOSE ZAHL 147 WERTPARAMETER 148 WHILE-ANWEISUNG
 149 WITH-ANWEISUNG 150 WORTSYMBOL 151 WRITE 152 WRITELN
 153 ZAHLEN 154 ZEICHEN 155 ZEICHENKETTENKONSTANTE 156 ZEIGERTYP
 157 ZEIGERVARIABLE 158 ZEILENSTRUKTUR 159 ZEILENWECHSEL 160 ZIFFER
 161 ZUFALLSZAHLEN 162 ZYKLUSANWEISUNGEN

SACHWORTVERZEICHNIS

ZU

PASCAL-PROGRAMMEN

PROGRAMM-NAMEN UND -NUMMERN

NR.	1	2	3	4	5	6	7	8	9	10
0	LEER	ZEIKETTE	TURM	STRING	STRINGS	PROVERB	PROGRAMM	KONSTANT	OPERATIO	KREIS01
10	ZEICHEN	INVENTUR	KREIS02	KUGEL	AUSGABE	PLANIST	INTEGER	KREIS03	LOGIK	FORDEMO
20	STRICHE	VIERECK	SCHNITT	HYPOTENU	ABSTAND	PRODUKTE	TETRAED	SKALPRO	KOPIE	ALFATYP
30	GLEICH	LAGE	QUALITAE	NAMEN	NULLTEIL	MASSE	ZWEIHOCH	HADDER	ZEICHTAB	SCHABRET
40	HILDAUS	SCHWING	BOOLE	WHILEDEM	FUNKTION	WERBUNG	SOLLWERT	POTENZEN	WERKZEUG	SCHWINGE
50	TAGE	AEGUIVAL	FELDER	VIERPOL	MATRIX	KREISTAB	ELEFUNKT	EIGENBAU	HAUS	MARKE
60	DREIECKE	FAKULTAE	STANDARD	ALFAWORT	VEKTOREN	REKURSIO	GAESTE	GGREKUR	CARDCOPY	DATENTYP
70	MITTEL	KOMPLEX	HUERER	KARTEI	VOLUMINA	KUENSTE	PLZAHL	ZEIGER	MATERIAL	LIFO
80	ANALYSEN	PROTANAL	OPTISCH	RECHTECK	FAECHER	SINUBILD	SINGRAF	FLAGGE	QUADRATE	PENROSE
90	KOERPER	ELLIPSEN	ELLIPSE	LIST	ZZAHLEN	INHALT	REGISTER	EFFEKT	EFFECT	SACHWORT
100	FORLOOP									

AKTUELLE PARAMETER 2 3 4 5 56 57 58 61 62 63 64 65 67 71 72 73 74 75 77 79 94
 95 97 98 99 100

ALFA 30 64 96
 ANFANGSWERT 3 20 21 22 23 28
 ANWEISUNG 2 3 4 5 12 13
 ANWEISUNGSTEIL 2 3 7 69
 ARITHMETISCHE FUNKTIONEN 24 25 36 61 63
 ARITHMETISCHE OPERATOREN 8 9 10 12 13 26 35 48 65
 AUFZAEHLUNGSTYP 51 65 70 75 76 101
 AUSDRUCK 8 9 10 12 13 27
 AUSGABE 2 3 12 13
 BEDINGTE ANWEISUNG 32 33 36 38 40 41 49 51 52 53 57 58 59 61 62 66 68
 BEZEICHNER 1 7 8 9 10 34 64
 BINAERER BAUM 81 82 100
 BLANK 3 6
 BLOCK 7 69
 BLOCKSTRUKTUR 7 8 9
 BOOLEAN-TYP 19 36 38 43 52 70 72 74
 BOOLESCHE OPERATOREN 19 38 43 52
 BOOLESCHER-AUSDRUCK 19 31 32 38 43 52
 BUCHSTABEN 2 11 46
 CASE-ANWEISUNG 51 57 75 76
 CASE-INDEX 76
 CHAR-TYP 11 16 39 40 46 70 73
 DATE 97
 DATENFELDLISTE 73 74 79 80 81 94 97 100
 DATENSATZKOMPONENTE 73 74 79 80 81 94 97 100
 DATENSATZTYPEN 70 72 73 74 75 76 79 80 81 94 97 100
 DATENTYPEN 70 74
 DIREKTIVEN 68 83 84 85 86 87 88 89 90 91 92 93

DRUCKFORMAT 15 16 17 18 19 20 21
 DYNAMISCHE VARIABLE 78 79 80 81 94 97 100
 EIN-/AUSGABE-PROZEDUREN 24 25 26 28 33
 EINFACHE ANWEISUNG 2 11 12 13 14 60
 EINFACHER TYP 11 12 13 30 70
 EINGABE 16 17 18
 ENDWERT 20 21 22 23 28 101
 ENTPACKEN 63
 ERGIBTANWEISUNG 26 30
 FALLKONSTANTENLISTE 51 57 75 76
 FELD 46 47 48 49 50 53 54 55 65 67 72 73 74 76
 FELDBREITE 15 18 21 22 23 26 27 46
 FELDTYPEN 46 47 48 49 50 53 54 55 65 67 70 73 74 76
 FESTER TEIL 72 73 74
 FILE 1 69 73 77
 FILEPUFFER 77
 FILETYPEN 70 77
 FOR-ANWEISUNG 20 21 22 23 27 28 29 30 33 37 38 39 40 41 42 43 45 45 46 47 49
 50 52 53 54 55 57 59 65 101
 FORMALE PARAMETER 56 57 58 61 62 63 64 65 67 68 71 72 73 74 75 77 79 81 82 83 84
 85 86 87 88 89 90 91 92 93 98 99 100
 FORWARD-DIREKTIVE 68
 FUNKTION 56 57 58 61 62 63 65 98 99
 FUNKTIONSAUFRUF 56 57 58 61 62 98 99
 FUNKTIONSDOKUMENTATION 56 57 58 61 62 98 99
 GEPACKTE DATENTYPEN 73 77
 GET 63 77
 GLOBAL 79 96
 GOTO-ANWEISUNG 60
 GRAFIK 3 6 15 21 22 25 27 27 30 38 42 50 59 59 83 84 85 86 87 88 89
 90 91 92 93 95
 GUELTIGKEITSBEREICH 61 62 63
 IF-ANWEISUNG 32 33 36 38 40 41 49 59 79
 INDEXTYP 46 47 48 49 50 53 54 55 65 67 73 74 76
 INDIZIERTE VARIABLE 46 47 48 49 50 53 54 55 65 67 73 74 76
 INTEGER-TYP 12 14 16 70
 KOMMENTAR 1 2 3 4 5 7 8 16
 KOMPONENTENTYP 73 74 76
 KOMPONENTENVARIABLE 46 47 48 49 50 53 54 55 65 67 73 74 76
 KONSTANTE 8 9 10 15
 KONSTANTENDEFINITION 8 9 10
 KONSTANTENDEFINITIONSTEIL 8 9 10 69
 KONVERTIERUNG 42 50
 LAUFVARIABLE 20 21 22 101
 LEERE ANWEISUNG 1 10 73
 LISTE 79 94 100
 LOKAL 61 62 64 66
 MARKEN 60 75
 MARKENDEKLARATIONSTEIL 60 69
 MENGE 51 76
 MENGENOPERATOREN 51
 MENGENTYP 51 70 76
 NAME 34
 ORDINALTYP 11 12 13 48 49
 OPERATOREN 8 9 10 17 18
 PACKEN 63 73 77 79
 PAD-ES 83 84 85 86 87 88 89 90 91 92 93
 PAGE 63 97
 PARAMETER 2 4 10
 PARAMETERLISTE 2 10 77
 PARAMETERUEBERGABE 96
 PROGRAMM 1 2 3 4 5 7
 PROGRAMMAUFBAU 2 3 7
 PROGRAMMKOPF 7
 PROGRAMMPARAMETER 1 2 3 4 5 7 64
 PROZEDUR 2 63 65 66 67 68 71 72 75 79 81 95
 PROZEDURAUFRUF 4 68 83 84 85 86 87 88 89 90 91 92 93 95
 PROZEDURKOPF 63 65 66 67 68 70
 PUFFERVARIABLE 77
 PUT 77
 QUELLTEXT 4
 READ 17 18 24 25 29 64
 READLN 16 17 18 24 25 29 64
 REAL-TYP 13 14 18 70
 REKURSIV 62 66 68
 REPEAT-ANWEISUNG 42
 RESET 63 77
 RESULTATTYP 19
 ROUTINE 45 63 64
 ROUTINEAUFRUF 45 63 64
 ROUTINEDEKLARATIONSTEIL 57 62 63 64 69
 ROUTINEKOEPE 57

```

5
2
24 25 39 42 44 45 50 63
[UNG 29 32 33 36 49 59
N 70
48 49 70
78
3 4
3 51 53 64 65 76 78 79 81 94 97 100 101
48 51 65 76 78 79 81 94 97 100
L 48 51 51 65 69 76 78 79 81 94 97 100
3 14 15 22 23 34
ON 11 12 13 14
ONSTEIL 11 12 13 14 69
68 71 72
29 53 101
7 69
EN 31 32
ABLE 11 12 13 14 15 16
ZAHL 8 9 10
STANTE 8 9
8 9 10
44 69 70 71 73 79 80
75 76 79
10 20 22 23 24 25 29 34 37
5 6 8 9 10 15 17 18 20 22 23 24 25 29 34
78
NTE 2 4 5 6 8 9 11 80 96
79 80 81 82 94 97 100
79 80 81 82 94 97 100
6 9 96
6 9
20 21 22 23 27 28 29 41 42 44 101
#####

```

R-Anweisung

NER AENDERUNG DES ENDWERTES EINER LAUFVARIABLEN IN EINER FOR-ANWEISUNG SIND ZU VER-

```

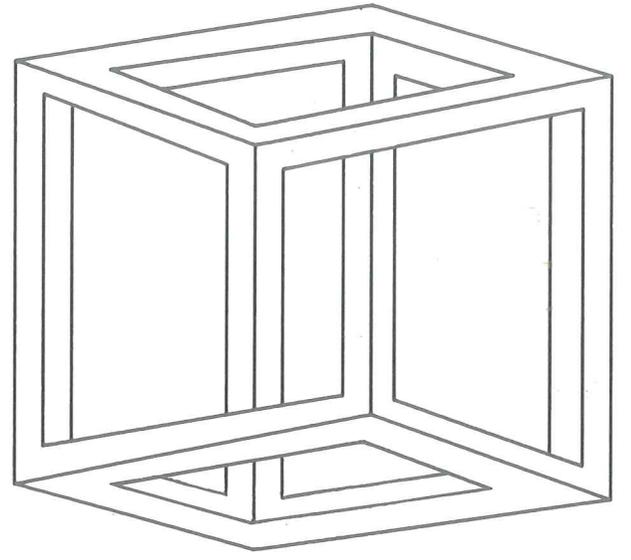
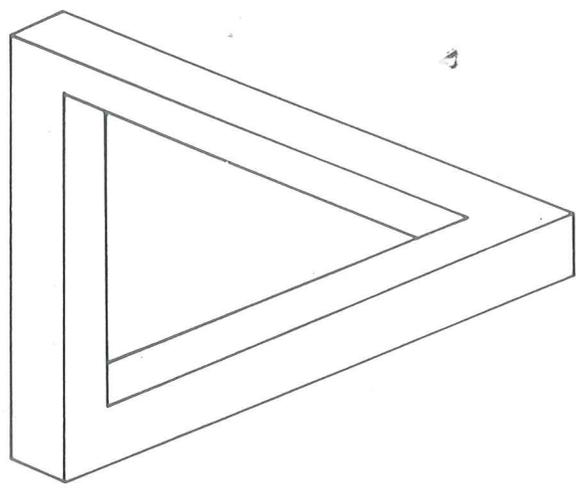
PUT);
NDWERTES EINER LAUFVARIABLEN *)
UFZAEHLUNGSTYP,ENDWERT *)
GRAU,RUT,ORANGE,GELB,GRUEN,
BLAU,VIOLETT,WEISZ);
VAR,ANFANG,ENDE:SPEKTRUM;
DER ABARBEITUNG:1:38);
TELN(' ');
2,'ENDE':12,'ZAEHLER':12,
14); WRITELN(' ');
:=VIOLETT; ZAEHLER:=0;
):12,ORD(ENDE):12,ZAEHLER:12);
G TO ENDE DO
+ 1;
ELN(' ');
NG):12,ORD(ENDE):12,ZAEHLER:12,
VAR):14);

```

PROTOKOLL DER ABARBEITUNG:

	ANFANG	ENDE	ZAEHLER	LAUFVAR
	1	7	0	
	1	4	1	1
	1	4	2	2
	1	4	3	3
	1	4	4	4
	1	4	5	5
	1	4	6	6
	1	4	7	7

aspekte blickpunkt



Beide Zeichnungen wurden mit PASCAL-Programmen erzeugt, die ihrerseits das Programmsystem PAD-ES vom VEB Carl Zeiss Jena nutzen, um damit Steuerprogramme für einen numerisch gesteuerten Zeichentisch zu erzeugen.

Der Versuch einer räumlichen Ordnung der beiden bereits im vorigen Jahrhundert bekannten Körper gelingt nicht und wird deshalb bereits nach kurzer Zeit eingestellt. Zugleich ergeben sich eine Reihe von Fragen:

- Wurde die Realität fehlerhaft modelliert?
- Wurden geeignete Modelle in den PASCAL-Programmen fehlerhaft abgebildet?
- Sind die Eingabedaten für die Programme oder die Beziehungen zwischen denselben fehlerhaft?
- Handelt es sich auch hier um Sinnestäuschungen, ähnlich wie bei der Zeichnung Optische Täuschung? (Zu den Beispielen „Penrosches Dreieck“ auf Seite 52 und „Ein nicht möglicher Körper“ auf Seite 53)

Hinweise zur Nutzung

Es wird davon ausgegangen, daß in der Regel von den einzelnen Nutzern auf Grund ihrer Vertrautheit mit der Programmierung von Computern, dem Umgang mit formalen Sprachen bzw. der Kenntnis der Sprache PASCAL jeweils unterschiedliche Zielstellungen angestrebt werden.

Beabsichtigt der Leser, die vorliegenden Aufgaben zur Unterstützung einer ersten Einführung in die höhere Programmiersprache PASCAL zu nutzen, so kann die gewählte Numerierung der Aufgaben, die aus dem VERZEICHNIS DER PROGRAMME (S.57) zu ersehen ist, sowohl einen ersten Einstieg bieten als auch ein roter Faden für ein systematisches Durcharbeiten der Aufgabensammlung sein. Die Numerierung beginnt bei den einfachsten Aufgabenstellungen und wird fortgesetzt mit Programmen, die zunehmend mehr bzw. leistungsstärkere Sprach-elemente umfassen und vielfach gegen-

Gleichzeitig wird in diesem Verzeichnis jedes der Programme kurz charakterisiert.

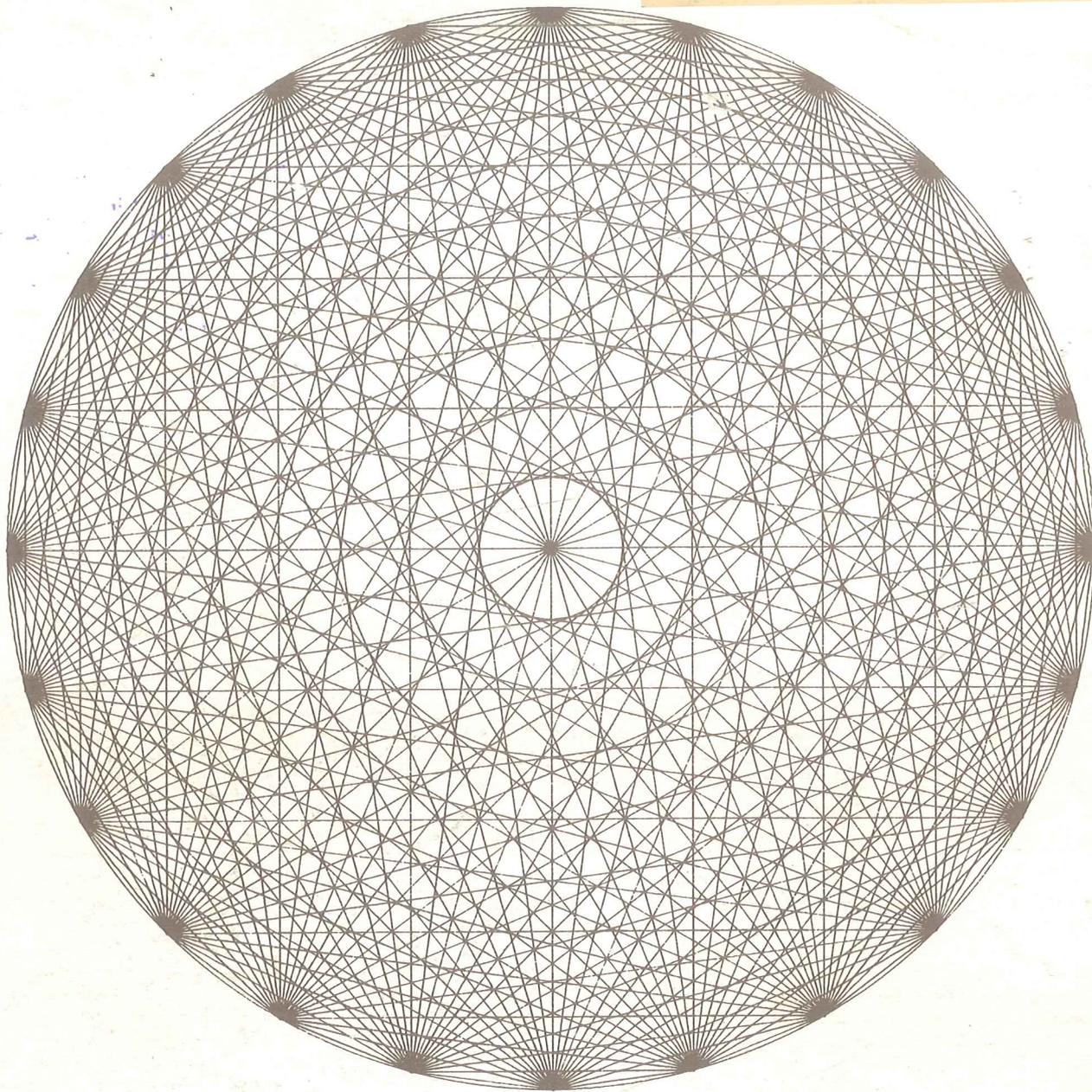
Eine weitere Hilfe liefert das PROGRAMM- UND AUFGABENREGISTER (S.59). Es verbindet die alphabetisch sortierten Namen der PASCAL-Programme mit den Bezeichnungen der zugehörigen Aufgabenstellungen. Gleichzeitig ist die bereits genannte laufende Nummer der Programme und Aufgaben angegeben, die beide identifiziert. Das Inhaltsverzeichnis auf der zweiten Umschlagseite listet die Bezeichnungen der Aufgabenstellungen auf und enthält neben der jeweils zugehörigen Seitennummer dieses Heftes auch die erwähnte laufende Nummer. Damit lassen sich nicht nur für jede Bezeichnung der Aufgabe, sondern auch für jeden Namen eines Programms bzw. auch für jede laufende Nummer die in diesem Heft erfolgte Platzierung der entsprechenden Aufgabenstellung und des Programms ermitteln.

SACHWORTVERZEICHNIS ZU PASCAL-PROGRAMMEN (S. 61/62), das seinerseits aus drei Teilen besteht:

- einer Zusammenstellung von Sachworten, deren Nutzung im genannten Register vorgesehen ist,
- einem Raster der Programm-Namen, das für eine gegebene laufende Nummer die Identifikation des zugeordneten Programms gestattet und
- einer alphabetisch geordneten Folge aller auftretenden Sachworte.

Nach jedem Sachwort sind die Nummern der Programme angeordnet, die mit dem betreffenden Sachwort im Zusammenhang stehen.

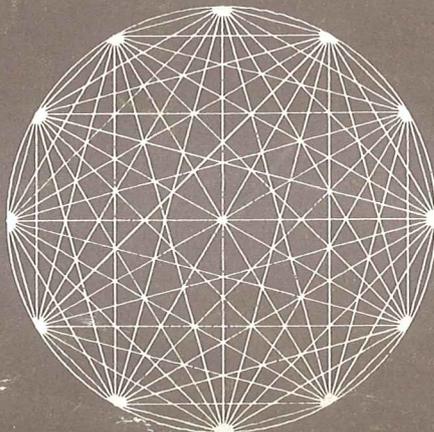
Damit wird der Zugang in der Regel zu einer Gruppe von Programmen und damit von Aufgaben erschlossen. Eine weitergehende Selektion unterstützt nun das genannte Verzeichnis der Programm-Namen. Gleichzeitig werden damit weitere Beziehungen zwischen den Programmen dargestellt, die nicht nur bei einer Einzel-



Auf einem Kreis ist eine endliche Anzahl von Punkten äquidistant angeordnet. Jeder Punkt soll mit einem jeden anderen Punkt verbunden werden. Zwei extreme Lösungsmöglichkeiten sind sofort erkennbar:

- die Verbindung durch den bereits genannten Kreis und
- die direkte Verbindung eines jeden Punktes mit einem jeden anderen.

Für die Gestaltung von Netzen (z. B. Datennetzen, Rechnernetzen, Kom-



munikationsnetzen) zeichnen sich nicht nur die bereits genannten, sondern noch weitere Strukturvarianten ab, so zum Beispiel Sternnetze.

Die Zeichnung läßt aber auch die Übergänge zwischen den bereits genannten extremen Lösungen erkennen und gibt zugleich Anlaß zu vielseitigen Überlegungen wie Länge der Verbindungen oder Belegungen bei Anrufen. (zu unserem Beispiel „Flagge“ auf Seite 51)