

dietz621

PASCAL-E
Benutzeranleitung

dietz 621

PASCAL-E

Benutzeranleitung

Heinrich Dietz
Solinger Straße 9
4330 Mülheim-Ruhr
Tel.: (0208) 44 34-1
Telex 856770

DIETZ Computer
SYSTEME

2-7901-03-079

Benutzeranleitung PASCAL

Inhaltsverzeichnis

Gesamt

Teil 1/3	Kurzbeschreibungen
	1. Allgemeines
	1.1 Kommandos
	1.2 Sprachelemente
	1.3 Standardimplementationen
	1.4 Sachwörterverzeichnis
Teil 2/3	Bedienung
	2. Allgemeines
	2.1 Kommandos
	2.2 Sprachelemente
	2.3 Sachwörterverzeichnis
Teil 3/3	Beispielsammlung

d/ietz621

PASCAL-E

Benutzeranleitung Teil 1/3
Kurzbeschreibung

Benutzeranleitung PASCAL Teil 1/3

Inhaltsverzeichnis

Kurzbeschreibungen

- 1.0 Allgemeines
- 1.1 Kommandos
 - 1.1.1 Alphabetische Liste
 - 1.1.2 Bedeutung der Kommandos
 - 1.1.3 Schreibweise der Kommandos
- 1.2 Sprachelemente
 - 1.2.1 Alphabetische Liste
 - 1.2.2 Aufbau eines Programms
 - 1.2.3 Aufbau des Vereinbarungsteils
 - 1.2.4 Aufbau des Ausführungsteils
- 1.3 Standardimplementationen
- 1.4 Sachwörterverzeichnis

Benutzeranleitung PASCAL Teil 1/3

1.1 Kommandos

1.1.1 Alphabetische Liste

Kommando	Seite
ALT, n, m	1.1-3
COMP, d	1.1-5
DEL, n, m	1.1-3
END	1.1-5
LIST, n, m	1.1-3
LOAD, u, f	1.1-2
READ, z, d	1.1-2
READ, z, u, f	1.1-2
RUN, d	1.1-5
SAVE, u, f	1.1-4
SCR	1.1-3

Benutzeranleitung PASCAL Teil 1/3

1.1.2 Bedeutung der Kommandos

1.1.2.1 Angabe der Quelldatei

READ, z, d
READ, z, u, f
LOAD, u, f

z = Zeilennummer

u = Unit (Ø, 1, 2 ..)

f = Filename (1..6 Zeichen)

d = Gerätenummer

READ, z, d Mit READ wird die Konsole oder die Platte als Quelldatei angegeben. Bei READ cr wird die aktuelle freie Zeilen-Nr. ausgegeben, danach kann die Eingabe über die Tastatur erfolgen. Eingabeende wird durch E CNTRL cr signalisiert.

READ, z, u, f Mit READ, z, u, f wird die Datei u, f hinter Zeile z geladen. Dieses Verfahren ist besonders beim nachträglichen Einfügen von Prozeduren und Funktionen von Vorteil.

LOAD, u, f Bei LOAD, u, f wird die Datei mit dem Namen f geladen.

Benutzeranleitung PASCAL Teil 1/3

1.1.2.2 Edition von Quelltext

LIST, n, m
DEL, n, m
ALT, n, m
SCR

n = 1. Zeilen-Nr.
m = letzte Zeilen-Nr.

- LIST, n, m Mit den Kommandos LIST wird das Symbolprogramm auf der Teilnehmerkonsole gelistet. Die Ausgabe des Programmes auf einen Drucker geschieht durch das Kommando COMP,d < cr>.
- DEL, n, m Das Löschen von Anweisungen geschieht einschl. der angegebenen Zeilen. Bei dem Kommando ALT
- ALT, n, m müssen die danach eingegebenen Zeilen mit E ^{CNTRL} <cr>beendet werden, wenn keine Einfügung von zusätzlichen Zeilen gewünscht werden.
- SCR Mit dem Kommando SCR wird der gesamte Speicherbereich des Benutzers gelöscht.

Beispiel:	LIST	Gesamtliste
	LIST, 10,10	Liste nur 10
	LIST, 23	Liste bis 23
	LIST, 35	Liste ab 35
	LIST, 12,30	Liste von 12 bis 30

Benutzeranleitung PASCAL Teil 1/3

1.1.2.3 Ablage von Quelltext

SAVE, u, f

u = Unit (Ø,1...)

f = Filename (1..6 Zeilen)

Mit dem Kommando SAVE wird auf dem Plattenlaufwerk (u) das Symbolprogramm unter dem Dateinamen (f) abgelegt. Defektmode ist sowohl für die Unitangabe (u) als auch für den Dateinamen (f) vorgesehen. Wird also nur SAVE als Kommando eingegeben wird der aktuelle Dateiname verwendet.

```
PASCAL
*SAVE
[UNIT],NAME: 0,PRNR01
```


Benutzeranleitung PASCAL Teil 1/3

1.1.2.4 Übersetzen von Quelltext

COMP,d

d = device (Ø,1,3...)

Mit dem Kommando COMPILE wird das Quellprogramm auf formale Richtigkeit überprüft und eine Liste auf dem angegebenen Gerät erstellt. Fehler werden mit einem Dreieck und der Fehlernummer gekennzeichnet. Eine verbale Beschreibung des Fehlers erfolgt am Ende des Programmes.

1.1.2.5 Starten des Programmes

RUN,d

Beim Starten des Programms erfolgt grundsätzlich eine Überprüfung des Programms, so daß dieses Kommando auch ohne ein vorheriges kompilieren eingegeben werden kann. Wird ein Gerät angegeben, so wird vor dem Programmstart die Symbolliste ausgegeben.

Benutzeranleitung PASCAL Teil 1/3

1.1.2.6 Rückkehr zum Betriebssystem

END

Falls sich in der Partition noch ein
Quellprogramm befindet, erfolgt die
Ausgabe:

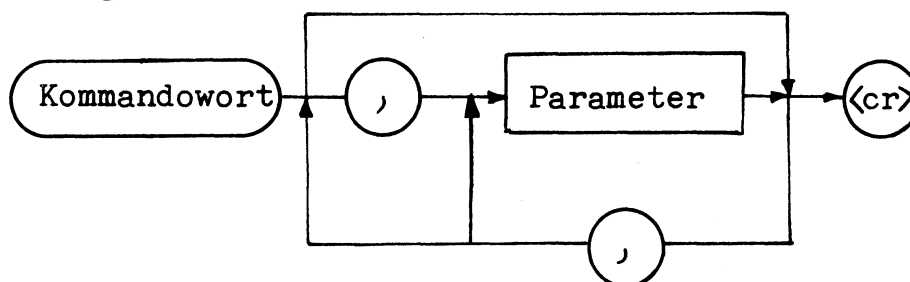
```

PASCAL
-END
ALERT: SOURCE NAME: PASCAL.P
PASCAL
/
```

Benutzeranleitung PASCAL Teil 1/3

1.1.3 Schreibweise der Kommandos

1.1.3.1 Allgemein



1.1.3.2

READ [, [z] [, d]]
READ [, [z] [, u] , f
LOAD [, u] , f

LIST [, [n] [, m]]

DEL , [n] [, m]

ALT , [n] [, m]

SAVE [, u] , f
COMP [, d]

RUN [,d]
SCR
END

Vom Kommandowort sind immer nur die ersten drei Zeichen von Bedeutung.

Angaben innerhalb der eckigen Klammern dürfen entfallen.

Benutzeranleitung PASCAL Teil 1/3

1.2.1 Alphabetische Liste

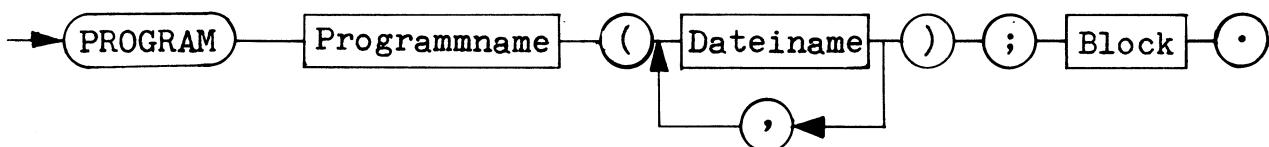
ARRAY	1.2-7
BEGIN/END	1.2-14
CASE/OF/OTHERS	1.2-16
CONST	1.2-4
FOR/TO/DOWNT0	1.2-17
FUNCTION	1.2-10
	1.2-22E
IF/THEN/ELSE	1.2-15
PROGRAM	1.2-2
PROCEDURE	1.2-11
	1.2-13
REPEAT/UNTIL	1.2-18
TYPE	1.2-5
VAR	1.2-7
WHILE/DO	1.2-18

Benutzeranleitung PASCAL Teil 1/3

1.2.2 Aufbau eines Programmes

1.2.2.1 Programm und Schnittstellendeklarationen

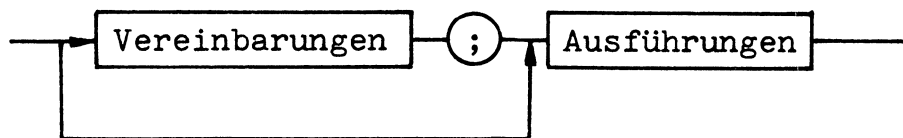
Ein PASCAL-Programm besteht aus dem Schlüsselwort 'PROGRAM', dem Programmnamen, der Schnittstellenfestlegung (Dateinamen) und dem 'Block'. Das Programm wird durch einen Punkt abgeschlossen.



Innerhalb des Bereichs 'Programmname' dürfen nur Buchstaben stehen.

1.2.2.2 Block

Ein Block besteht aus dem Vereinbarungsteil und dem Ausführungsteil. Der Ausführungsteil beginnt mit dem Schlüsselwort 'BEGIN' und endet mit dem Schlüsselwort 'END'. Das Schlüsselwort 'END' ist logisches Ende des Programmes.

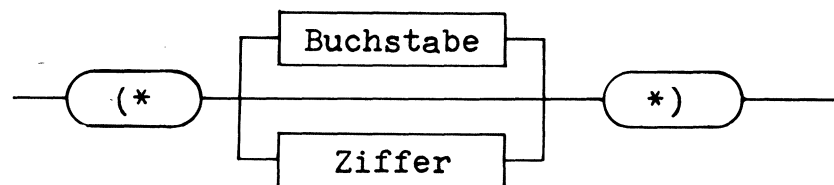


Ein Block muß nicht unbedingt einen Vereinbarungsteil haben. Wenn kein Vereinbarungsteil vorhanden ist, können im Ausführungsteil nur konstante Objekte verwendet werden.

Benutzeranleitung PASCAL Teil 1/3

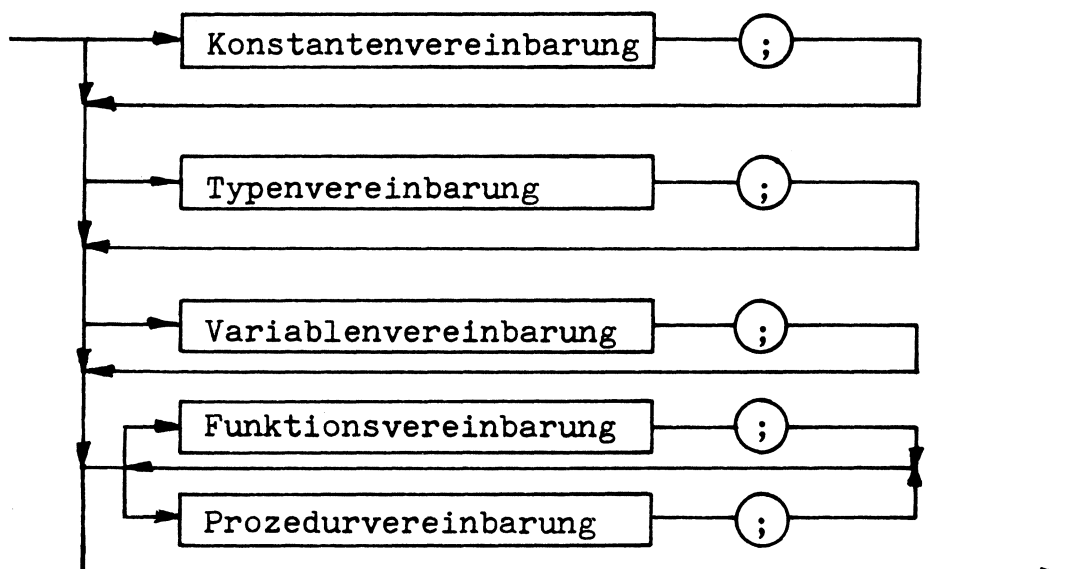
1.2.2.3 Programmerläuterungen

Kommentar kann an beliebigen Stellen des Programmes eingefügt werden. Zur Abgrenzung des Kommentars werden die Zeichen Klammer und Stern verwendet.



1.2.3 Aufbau des Vereinbarungsteils

Im Vereinbarungsteil können in bestimmter Reihenfolge Konstanten, Typen, Variable, Funktionen und Prozeduren dem System bekannt gegeben werden. Die Reihenfolge der Vereinbarungen ergibt sich aus folgendem Syntaxdiagramm.



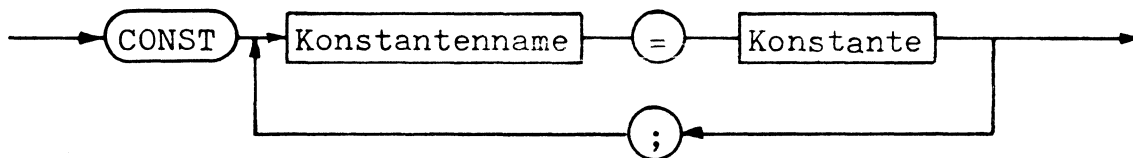
Benutzeranleitung PASCAL Teil 1/3

Unter Vereinbarungen versteht man die eindeutige Zuordnung von Namen zu Konstanten, Typen, Variablen, Funktionen und Prozeduren, wobei diese Zuordnung einen bestimmten Gültigkeitsbereich hat.

Gültig ist eine Vereinbarung nur in dem Block, in dem sie angegeben wurde und in allen Blöcken die intern in diesem Block enthalten sind, sofern nicht in einem der internen Blöcke - was erlaubt ist - der gleiche Name verwendet wird. In dem Fall gilt die neue Vereinbarung für diesen Block und alle weiteren, die wiederum in diesem enthalten sind.

1.2.3.1 Vereinbarung von Konstanten

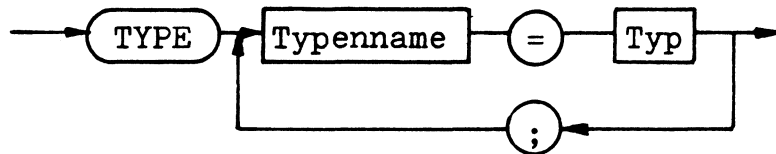
Die Vereinbarung von Konstanten erlaubt es, einem Namen einen festen Wert zuzuweisen. Der Typ ergibt sich aus der Konstante. Ein Konstantenname darf nicht auf der linken Seite einer Zuweisung verwendet werden.



Ein vereinbarter Konstantenname darf in einer danach folgenden Konstantenvereinbarung verwendet werden.

1.2.3.2 Vereinbarung von Typen

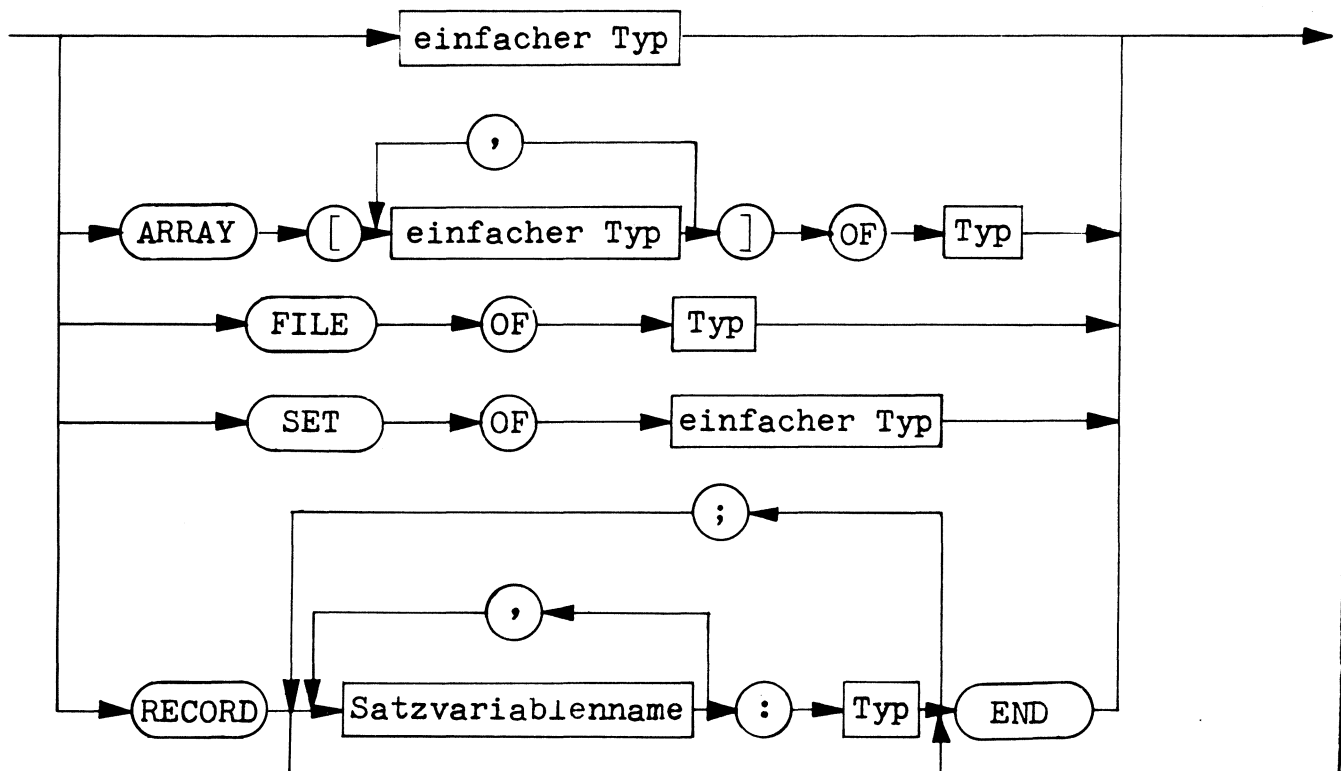
Diese Vereinbarung von Typen erlaubt es, einen Namen einen festen Datentyp zuzuordnen. Dieser Name darf in weiteren Typvereinbarungen bzw. bei der Vereinbarung von Variablen, Funktionen und in formalen Parameterlisten als Typname verwendet werden.



Es können damit einfache oder strukturierte Typen vereinbart werden.

Einfache Typen sind Skalare, Standard und Teilbereiche, strukturierte Typen sind Bereiche, Mengen, Sätze und Dateien.

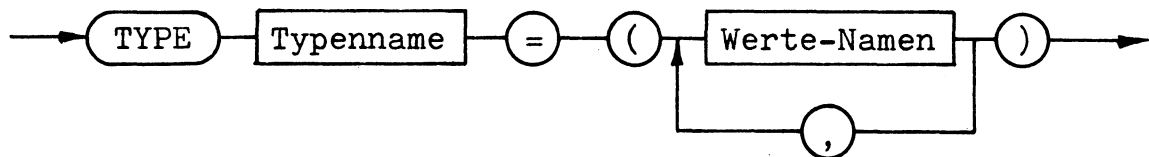
TYP



Benutzeranleitung PASCAL Teil 1/3

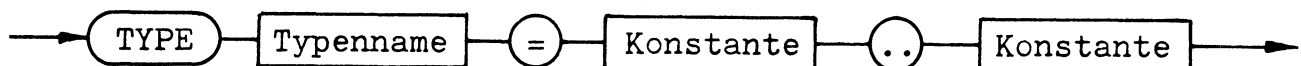
1.2.3.3 Vereinbarung von skalaren Typen

Durch die Vereinbarung wird einem Namen(Typname) eine Wertmenge bestehend aus Namen zugewiesen. Im Ausführungsteil kann einer Variablen, die von diesem Typ ist, ein Wert aus der Wertemenge zugewiesen werden. Durch die Aufzählung der Namen in der Wertemenge ist eine Ordnung gegeben, so daß die Standardfunktionen 'PRED' und 'SUCC' (Vorgänger bzw. Nachfolger) verwendet werden können. Ebenso ist es auch möglich mit einem skalaren Typ vereinbarte Variablen als Schleifenzähler zu verwenden.



1.2.3.4 Vereinbarung von Teilbereichen (Subrange)

Es ist möglich, mit Hilfe der zählbaren Typen Integer, Boolean, Char und Skalar neue Typen mit eingeschränkter Wertemenge zu vereinbaren. Variable von diesen Typen dürfen nur Werte innerhalb des vereinbarten Intervalls annehmen. Es darf dabei nur ein Intervall vorgegeben werden.

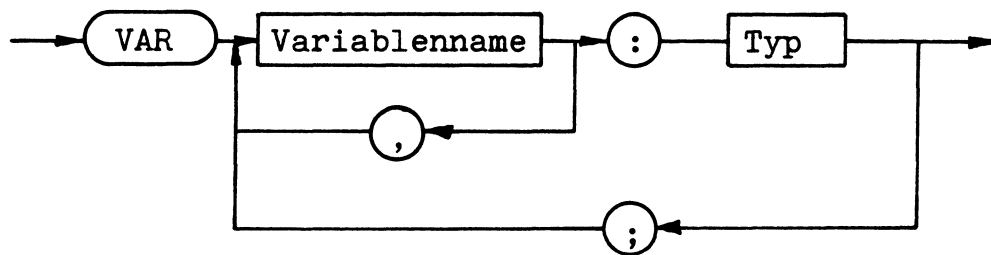


Benutzeranleitung PASCAL Teil 1/3

1.2.3.5 Vereinbarung von Variablen

- Vereinbarung von einfachen Variablen

Variablen sind veränderbare Objekte, die durch einen Namen angesprochen werden können. Variablen müssen deklariert werden. Dabei wird auch der Typ festgelegt.



- Vereinbarung von strukturierten Variablen

Bei strukturierten Variablen wird bei der Deklaration neben dem Typ auch die Struktur festgelegt.

- Bereichsvariable (ARRAY)

Eine ARRAY ist eine Zusammenfassung von n Elementen des selben Typs zu einer Einheit. Die einzelnen Elemente eines ARRAY's können über einen Index angesprochen werden. Ein ARRAY wird beschrieben durch den Typ des Elementes und den Typ des Indexes. Der Typ des Indexes muß zählbar sein (z.B. Integer, Character). Bei Integer ist allerdings nur ein Teilbereich (Subrange) möglich.

Benutzeranleitung PASCAL Teil 1/3

- Satzvariable (RECORD)

Ein 'RECORD' ist die Zusammenfassung von Elementen verschiedenen Typs zu einem Verbund. Die Zugriff auf die Feinstruktur (Element) erfolgt über Selektoren. Der Typ eines RECORD-Elementes ist beliebig, darf aber nicht vom Typ 'FILE' sein.

- Mengenvariable

Eine Variable vom Typ Menge besteht aus maximal 64 Elementen. Ein Mengentyp beschreibt die Potenzmenge seines Basistyps, d. h. alle im gegebenen Wertebereich möglichen Teilmengen. Basistyp einer Menge ist immer ein zählbarer Typ.

Für die Wertebereiche S der verschiedenen Basistypen gilt:

- Basistyp	- Wertmenge
INTEGER	$\emptyset \leq S \leq 63$
CHAR	"␣" $\leq S \leq$ "↑"
Skalar Typ	Anzahl der Namen ≤ 64

Benutzeranleitung PASCAL Teil 1/3

- Bei skalaren Typen ist die Reihenfolge der Aufzählung maßgeblich für die Ordinalzahl eines Elementes. Für Variable vom Typ Menge sind folgende Operatoren definiert.

+, OR	Vereinigung
-	Differenz
*, AND	Durchschnitt
<, < =	Vergleich auf Teilmenge
>, > =	Vergleich auf Obermenge
=, <>	Vergleich auf gleich/ungleich
IN	Vergleich auf Element von

Alle Operatoren sind entsprechend der Definition der Mengenalgebra implementiert.

Benutzeranleitung PASCAL Teil 1/3

1.2.3.6 Vereinbarung von Funktionen

Eine Funktionsdeklaration erklärt ein Programmteil zur Funktion und ordnet ihr einen Namen zu. Dieser ermöglicht den späteren Aufruf der Funktion.

Die Funktion besteht aus Funktionskopf (FUNCTION, Funktionsname, Parameterliste, Typname) und dem Funktionskörper (Block).

Die Parameterliste enthält - sofern sie nicht leer ist - die formalen Parameter, denen bei Funktionsaufruf aktuelle Werte zugewiesen werden.

Referenz-Parameter werden mit dem Schlüsselwort 'VAR' gekennzeichnet, Funktionsparameter mit 'FUNCTION' und Prozedurparameter mit 'PROCEDURE'. Alle nicht durch eines dieser Schlüsselwörter gekennzeichneten Parameter sind Value-Parameter.

Der Typname im Funktionskopf beschreibt den Ergebnistyp der Funktion; Ergebnistyp muß ein einfacher Typ oder ein Mengentyp sein.

Innerhalb des Funktionskörpers muß mindestens einmal eine Wertzuweisung auf den Funktionsnamen erfolgen. Das Ergebnis dieser Zuweisung wird nach Beendigung der Funktion an die aufrufende Stelle übergeben.

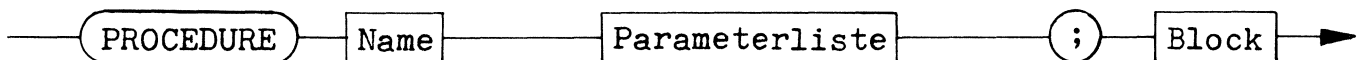
Eine Funktion kann sich rekursiv aufrufen, d.h. innerhalb einer Funktion kann ein Funktionsaufruf mit deren Funktionsname erfolgen.

Benutzeranleitung PASCAL Teil 1/3

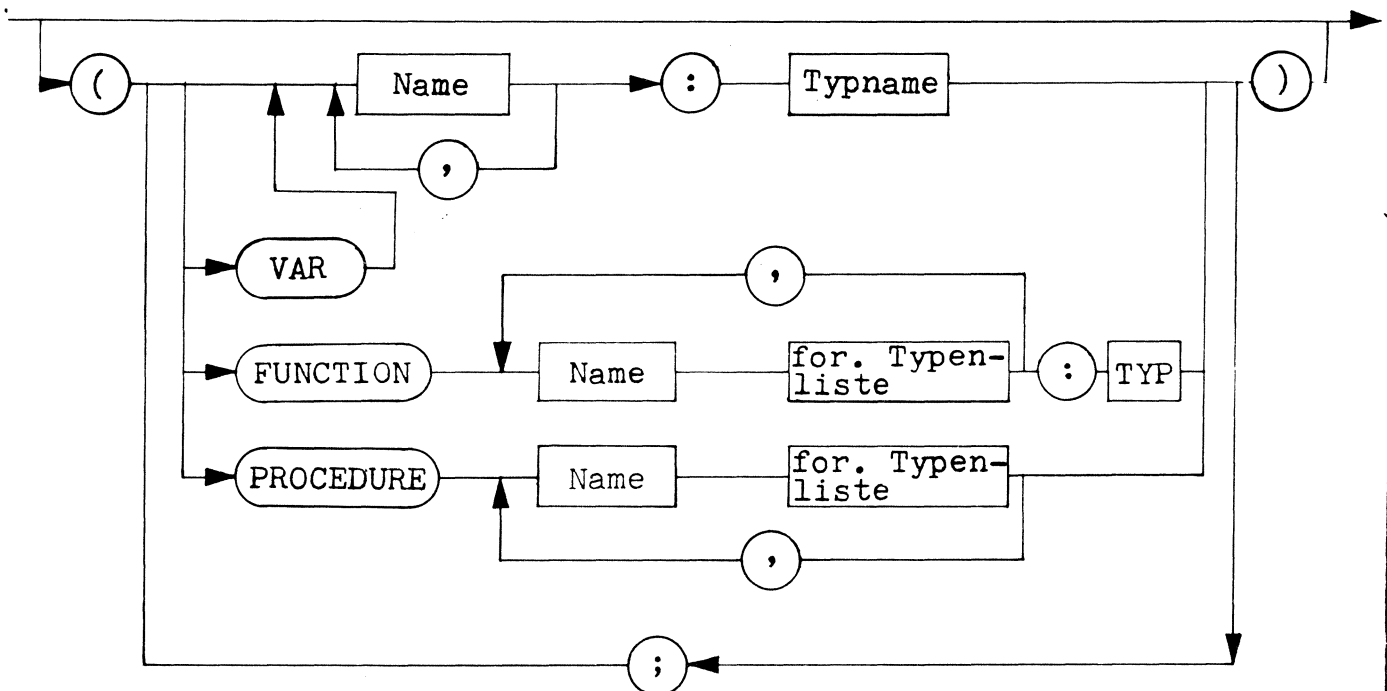
1.2.3.7 Vereinbarung von Prozeduren

Eine Prozedurdeklaration erklärt einen Programmteil zur Prozedur und ordnet ihm einen Namen zu; dieser Name ermöglicht den späteren Aufruf der Prozedur.

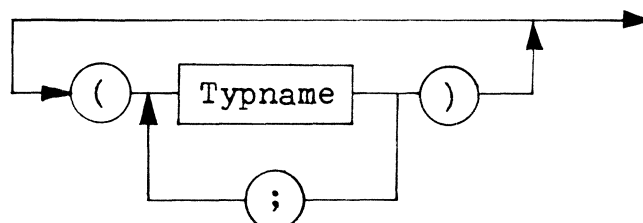
Die Prozedurdeklaration besteht aus dem Prozedurkopf (PROCEDURE, Prozedurname, Parameterliste) und dem Prozedurkörper (Block). Bezüglich Parametern und Rekursivität einer Prozedur gelten die selben Angaben wie für eine Funktion.



Parameterliste



Formale Typenliste



Benutzeranleitung PASCAL Teil 1/3

1.2.4 Ausführungsteil

Der Ausführungsteil wird als Verbundanweisung geschrieben, d. h. ein PASCAL-Programm besteht nur aus einer einzigen Anweisung. Eine Verbundanweisung ist eine Zusammenfassung von beliebig vielen Anweisungen durch die Schlüsselworte BEGIN und END. Die einzelnen Anweisungen sind durch ein Semikolon von einander zu trennen. PASCAL unterscheidet einfache und strukturierte Anweisungen.

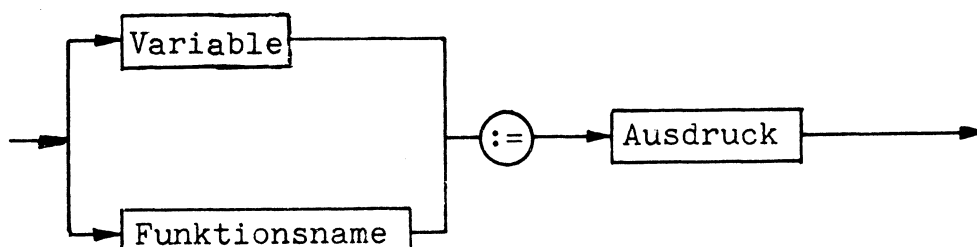
1.2.4.1 Einfache Anweisungen

Einfache Anweisungen sind Zuweisungen und Prozeduraufrufe.

- Zuweisungen

In einer Zuweisung wird einer Variablen oder einer Funktion ein Wert zugeordnet, der aus einem Ausdruck bestimmt wird. Variable bzw. Funktion und Ausdruck müssen vom selben Typ sein. Ausnahmen, die ebenfalls erlaubt sind:

- Variable bzw. Funktionen vom Typ REAL und Ausdruck vom Typ Integer
- Ausdruck ist Teilbereich über dem Typ der Variablen und umgekehrt.



Benutzeranleitung PASCAL Teil 1/3

- Prozeduraufrufe

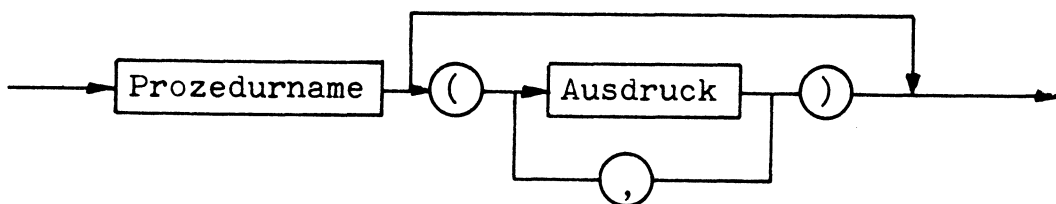
Die Nennung eines deklarierten Prozedurnamens als Anweisung veranlaßt die Ausführung der genannten Prozedur. Ist die Parameterliste der aufgerufenen Prozedur nicht leer, so muß ein Prozeduraufruf über eine Liste jedem formalen Parameter ein aktueller Parameter zugeordnet werden. Die Zuordnung geschieht dabei über die Position der formalen und aktuellen Parameter innerhalb ihre Listen. Folgende Bedingungen müssen bei der Zuordnung eingehalten werden.

Auf formale Parameter, die von einem Strukturtyp sind, dürfen nur aktuelle Parameter zugeordnet werden, die typidentisch sind.

Auf alle anderen formalen Parameter, die Value-, Referenz- oder Funktionsparameter sind, dürfen nur typkompatible aktuelle Parameter zugeordnet werden, auf Value-Parameter vom Typ Real auch aktuelle Parameter vom Typ Integer oder Integer-subrange.

Auf formale Referenzparameter dürfen nur Variable zugeordnet werden.

Funktions- und Prozedurparameter werden ohne Parameter übergeben.

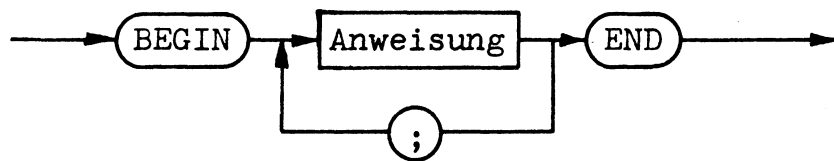


1.2.4.2 Strukturierte Anweisungen

Zu den strukturierten Anweisungen zählen die Verbundanweisung, die Verzweigungsanweisungen (IF, CASE) und die Wiederholungsanweisungen (FOR, WHILE, REPEAT).

- Verbundanweisung

Die Verbundanweisung faßt Anweisungen zwischen den Schlüsselwörtern BEGIN und END zusammen. Die einzelnen Anweisungen werden durch Semikolon voneinander getrennt.

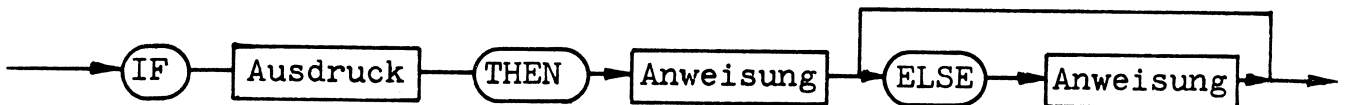


Benutzeranleitung PASCAL Teil 1/3

- Verzweigungsanweisungen

- IF Anweisung

Entweder wird die Anweisung hinter dem Schlüsselwort 'THEN' ausgeführt - wenn der Wert des Ausdrucks (boolsch) hinter 'IF' den Wert 'TRUE' ergibt - oder aber es wird - wenn der Ausdruck den Wert 'FALSE' ergibt - die Anweisung hinter 'ELSE' ausgeführt. Fehlt in diesem Fall die Angabe von 'ELSE', wird das Programm mit der Anweisung, die der IF-Anweisung logisch folgt, fortgesetzt.



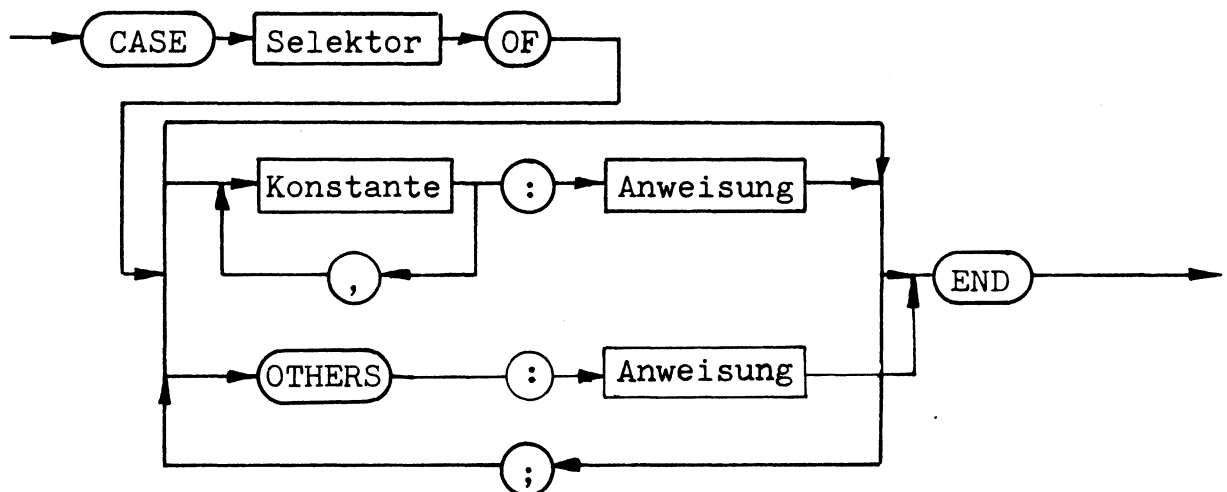
Benutzeranleitung PASCAL Teil 1/3

- Verzweigungsanweisungen

- CASE Anweisung

Es wird genau die Anweisung ausgeführt, deren Markenkonstante den gleichen Wert hat, wie der Ausdruck, der dem Schlüsselwort 'CASE' folgt.

Markenkonstanten und Selektor müssen vom gleichen zählbaren Typ oder einem Teilbereichstyp davon sein. Existiert keine Markenkonstante mit dem Wert des Selektors, dann wird die Anweisung hinter der Marke 'OTHERS' ausgeführt. Ist 'OTHERS' nicht explizit angegeben, so wird sie vom Compiler intern, gefolgt von einer leeren Anweisung, eingesetzt. Die Marke 'OTHERS' darf nur als letzte Marke innerhalb einer CASE-Anweisung aufgeführt werden.



- Wiederholungen

- FOR Anweisung

Die Anweisung hinter 'DO' wird n-Mal ausgeführt.

n -Ausführung bei

'TO' falls Endwert \geq Anfangswert

'DOWNTO' falls Anfangswert \geq Endwert

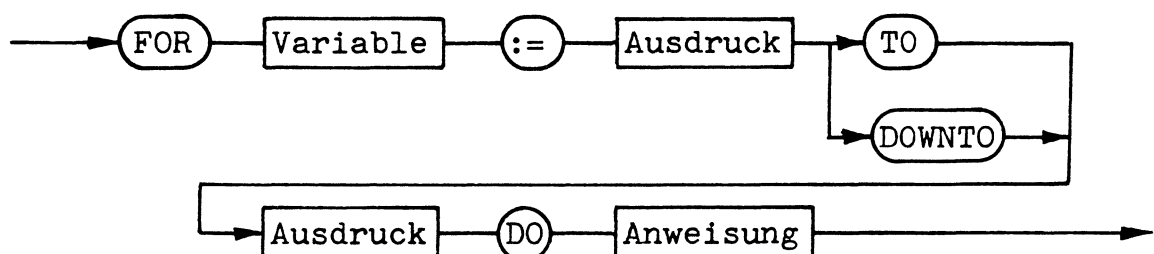
$$n = (\text{Endwert} - \text{Anfangswert}) + 1 \quad \text{bzw.} \\ (\text{Anfangswert} - \text{Endwert}) + 1$$

Dies gilt, wenn es sich um Integer-Werte handelt, bei allen anderen erlaubten Typen gilt sinngemäß ORD (Anfangswert) bzw. ORD (Endwert).

Kontrollvariable, Anfangs- und Endwert müssen vom gleichen, einfachen Typ (außer Real) sein.

Anfangs- und Endwert werden bei Eintritt in die FOR- Anweisung bestimmt, liegen also nach Eintritt in die FOR-Anweisung fest. In der Anweisung hinter 'DO' können sie deshalb nicht mehr beeinflusst werden.

Die Kontrollvariable darf innerhalb dieser Anweisung in allen Ausdrücken verwendet werden; sie hat den Wert Anfangswert + M bei Integertypen, bzw. den Wert, der die Ordinalzahl ORD (Anfangswert) + M besitzt, bei allen anderen erlaubten Typen. M bedeutet die aktuelle Anzahl an Wiederholungen.

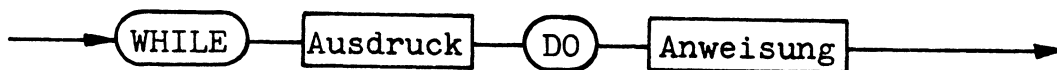


Benutzeranleitung PASCAL Teil 1/3

- Wiederholungen

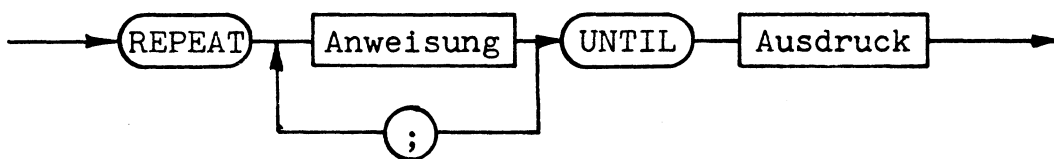
- WHILE Anweisung

Solange der Wert des Ausdrucks, der dem Schlüsselwort 'WHILE' folgt, den Wert 'TRUE' hat, wird die Anweisung hinter dem Schlüsselwort 'DO' ausgeführt. Besitzt der Ausdruck schon zu Beginn den Wert 'FALSE' wird die Anweisung nicht ausgeführt.



- REPEAT Anweisung

Die Anweisung zwischen den Schlüsselwörtern 'REPEAT' und 'UNTIL' wird solange ausgeführt, bis der boolsche Ausdruck, der 'UNTIL' folgt, den Wert 'TRUE' annimmt. Ist dieser Ausdruck schon zu Beginn 'TRUE', wird die Anweisung einmal ausgeführt.



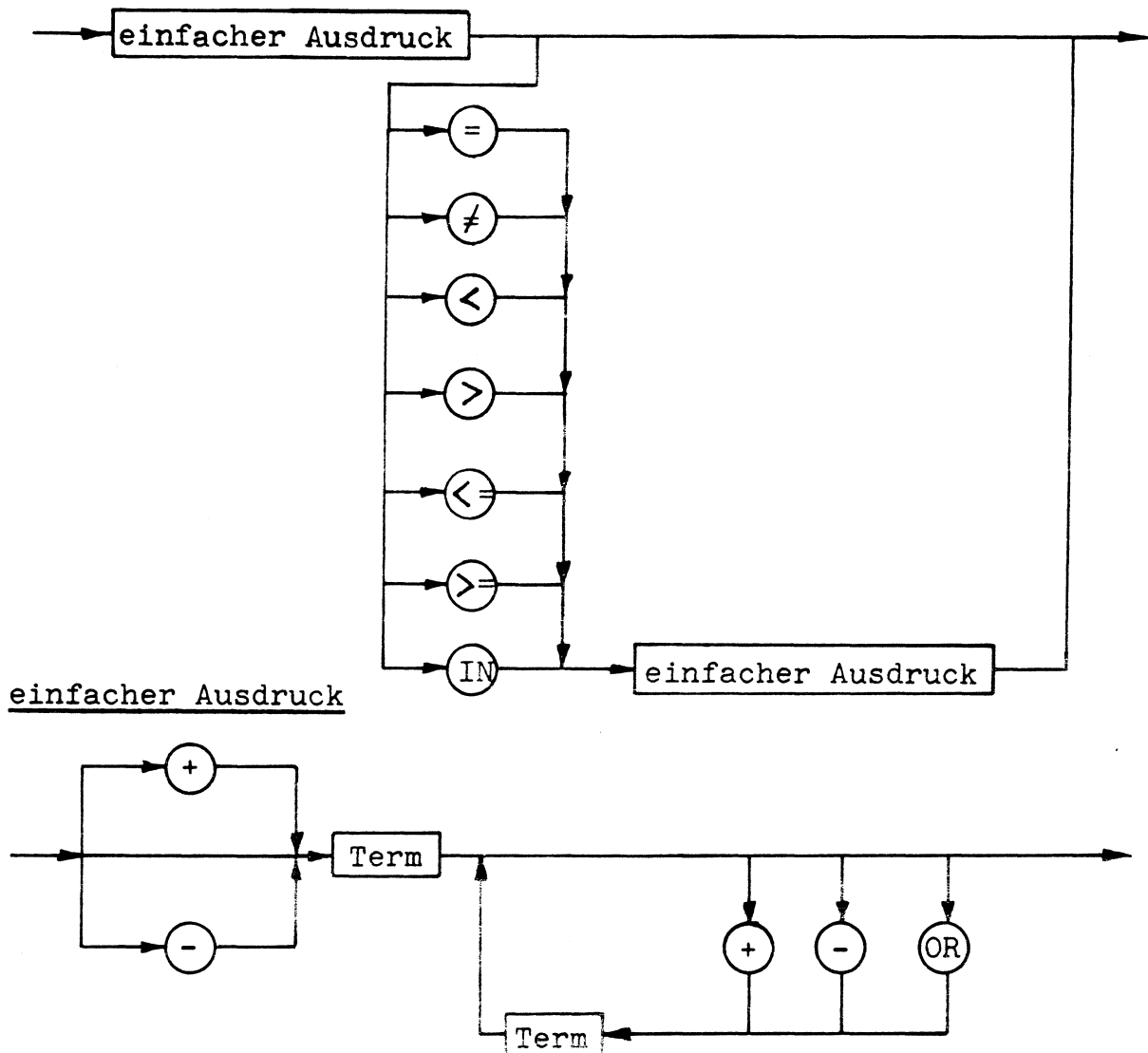
Benutzeranleitung PASCAL Teil 1/3

1.2.4.3 Ausdrücke

Ausdrücke werden aus Konstanten, Variablen, Operatoren und Funktionsaufrufen gebildet. Durch die Syntax der Ausdrücke wird auch die Priorität der einzelnen Operatoren bestimmt. Es ergibt sich folgende Reihenfolge:

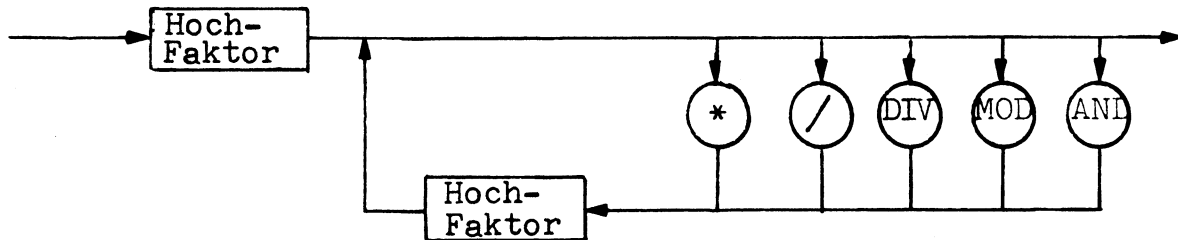
Höchste Priorität hat die Negation, gefolgt von der Exponentiation, den Multiplikations-, Additions- und den Relationsoperatoren. Zu beachten ist, daß diese Reihenfolge für alle Typen gilt, so daß die boolschen Operatoren auf der gleichen Stufe stehen, wie die arithmetischen.

Nach der Ausführung eines Ausdrucks ergibt sich als Ergebnis ein Wert, dessen Typ aus dem Ausdruck bestimmbar ist.

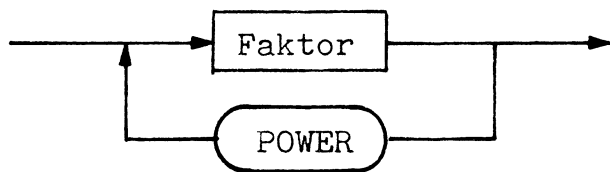
Ausdruck


Benutzeranleitung PASCAL Teil 1/3

Term

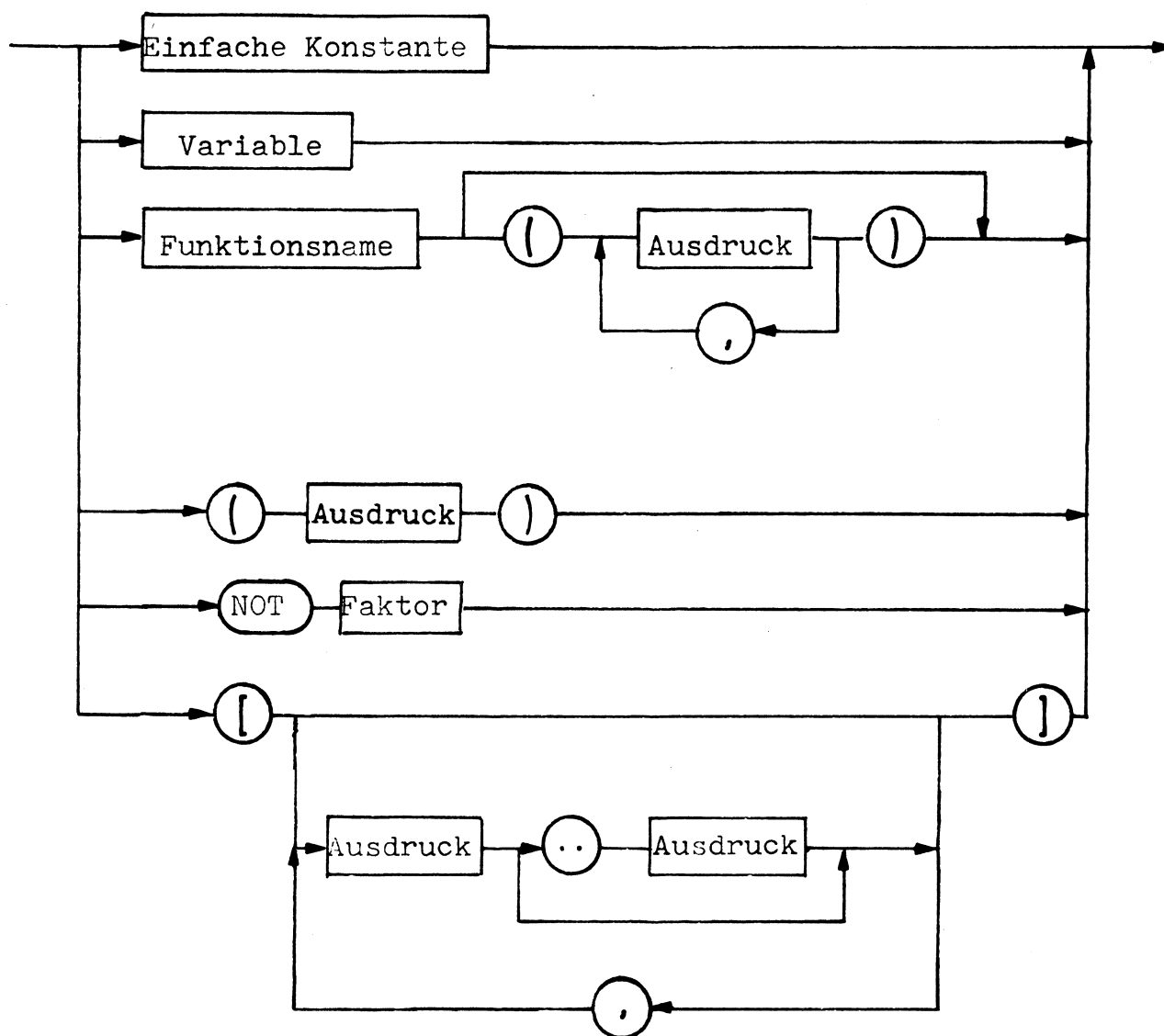


Hochfaktor



Benutzeranleitung PASCAL Teil 1/3

Faktor



1.2.4.4 Funktionsaufrufe

Durch Nennung eines deklarierten Funktionsnamens innerhalb eines Ausdrucks wird die Ausführung der genannten Funktion veranlaßt und deren Resultat anstelle des Aufrufs in den Ausdruck eingesetzt.

Ist die Parameterliste der aufgerufenen Funktion nicht leer, so müssen an deren formale Parameter aktuelle Parameter übergeben werden. Die Zuordnung geschieht dabei über die Position der formalen und aktuellen Parameter innerhalb ihrer Listen.

Folgende Bedingungen müssen bei der Zuordnung eingehalten werden:

Auf formale Parameter, die von einem Strukturtyp sind, dürfen nur aktuelle Parameter zugeordnet werden, die typidentisch sind.

Auf alle anderen formalen Parameter, die Value-, Referenz- oder Funktionsparameter sind, dürfen nur typkompatible aktuelle Parameter zugeordnet werden, auf Value-Parameter vom Typ Real auch aktuelle Parameter vom Typ Integer oder Integer-subrange.

Auf formale Referenzparameter dürfen nur Variable zugeordnet werden.

Funktions- und Prozedurparameter werden ohne Parameter übergeben.

Benutzeranleitung PASCAL Teil 1/3

1.3 Standard-Implementationen

1.3.1 Standard-Typen

Folgender Speicherplatz wird belegt:

Variablentyp	Speicherplatz	Wertebereich
INTEGER	4 Byte	$-2^{31} \leq n < 2^{31}$
REAL	8 Byte	$-2^{55} \leq n < 2^{55}$
BOOLEAN	1 Byte	'TRUE' und 'FALSE'
CHAR	1 Byte	alle ASCII-Zeichen
SKALAR	4 Byte	1 bis n Namen
SUBRANGE	wie Basis- typ	laut Typdefinition
SET	8 Byte	0 bis 64 Elemente
ARRAY	} Summe aller Komponenten	} Dem Komponenten- typ entsprechend
RECORD		

Benutzeranleitung PASCAL Teil 1/3

1.3.2 Operatoren

Operator	Typ 1. Op.	Typ 2. Op.	Typ Ergebnis
NOT	BOOLEAN	===	BOOLEAN
POWER	INTEGER	INTEGER	INTEGER
	INTEGER	REAL	REAL
	REAL	INTEGER	REAL
	REAL	REAL	REAL

- Multiplikationsoperatoren

*	INTEGER	INTEGER	INTEGER
	INT.REAL	INT.REAL	REAL
	MENGE	MENGE	MENGE
/	INT.REAL	INT.REAL	REAL
DIV	INTEGER	INTEGER	INTEGER
MOD	INTEGER	INTEGER	INTEGER
AND	BOOLEAN	BOOLEAN	BOOLEAN
	SET	SET	SET

- Additionsoperatoren

+	INTEGER	INTEGER	INTEGER
	INT.REAL	INT.REAL	REAL
-	WIE PLUS		
OR	BOOLEAN	BOOLEAN	BOOLEAN
	SET	SET	SET

- Vergleichsoperatoren

= <>	Bel.Typ T1	Typ T1	BOOLEAN
< > <=>=	Einfacher Typ	Einfacher Typ	BOOLEAN
	Mengentyp	Mengentyp	BOOLEAN
'IN'	Basistyp einer Menge	Menge	BOOLEAN

Benutzeranleitung PASCAL Teil 1/3

1.3.3 Standardfunktionen

Standardfunktionen sind bei der Implementation vordeclarierte Funktionen.

- Arithmetische Funktionen:

ABS(a)	: Berechnet den Absolutwert des Ausdrucks (a). Der Wert des Ausdruck muß vom Typ Real oder Integer sein. Der Resultattyp ist gleich dem des Ausdrucks.	
SQR(a)	: Berechnet a hoch 2. Der Wert des Ausdrucks muß vom Typ Real oder Integer sein. Der Resultattyp ist gleich dem von a.	
SQRT(a)	: Quadratwurzel von a	
SIN(a)	: Sinus von a	
COS(a)	: Cosinus von a	
ARCTAN(a)	: Arcustangans von a	Typ von a: Real oder Integer
EXP(a)	: e hoch a	Resultattyp: Real
LN(a)	: natürlicher Log. von a	

- Boolsche Funktionen:

ODD(a)	: Typ des Ausdrucks (a) muß integer sein, Resultattyp ist BOOLEAN. Ist der Wert von (a) ungerade, dann ist das Resultat TRUE, sonst FALSE.
EOF(f)	: Ist das Ende der Datei (f) erreicht, (Komponentenmarkierung zeigt hinter die letzte Komponente), dann ist EOF(f) TRUE, sonst FALSE.
EOLN(f)	: Ist das Ende der Datei (f) (Komponententyp 'CHAR') erreicht, ist der Wert von EOLN(f) TRUE, sonst FALSE.

1.3.4 Umwandlungsfunktionen:

- TRUNC(a) : a ist vom Typ Real; Ergebnis ist die
nächstniedere Ganzzahl zu a
- ROUND(a) : a ist vom Typ Real; Ergebnis ist die
nächstliegende Ganzzahl zu a
- ORD(a) : a muß vom Typ Skalar, Boolean oder
Charakter sein; das Resultat vom Typ
Integer ist die Ordinalzahl (Ordnungs-
zahl) für den Wert innerhalb der Werte-
menge, die durch den Typ von a definiert
ist.
- CHR(a) : a muß vom Typ Integer sein, das Resultat
vom Typ Character ist das Zeichen aus der
Wertemenge aller Zeichen, dessen Ordinal-
zahl der Wert von a darstellt.

Weitere Standardfunktionen:

- SUCC(a) : a muß von einem einfachen Typ sein außer
Real. Das Resultat ist der Nachfolger des
Wertes aus der Wertemenge, die durch
den Typ von a definiert ist, sofern ein
Nachfolger vorhanden ist.
- PRED(a) : a muß von einem einfachen Typ sein außer
Real. Das Resultat ist der Vorgänger des
Wertes aus der Wertemenge, die durch
den Typ von a definiert ist.

a = Ausdruck

DEV(d) : Umschaltung auf ein anderes Ein-/Ausgabegerät

d = device (Gerät)

Die Anwahl der Device-Nr. d ist statisch. Die
selektierte Device-Nr. gilt auch für die An-
weisungen 'READ' und 'READLN'. Das Umschalten
der Device-Nr. ist nur mit den Anweisungen
'WRITE' und 'WRITELN' möglich.

Benutzeranleitung PASCAL Teil 1/3

1.4

Sachwörterverzeichnis

ABS(a)	1.3-3
ALT,n,m	1.1-3
AND	1.2-9
AND	1.3-2
ARCTAN(a)	1.3-3
ARRAY	1.2-5
ARRAY	1.2-7
ARRAY	1.3-1
Ausdruck	1.2-19
Ausführungen	1.2-2
BLOCK	1.2-2
BOOLEAN	1.3-1
Buchstabe	1.2-3
CASE/OF/OTHERS	1.2-16
CHAR	1.3-1
CHR(a)	1.3-4
COMP,d	1.1-5
CONST	1.2-4
COS(a)	1.3-3
Dateiname	1.2-2
DEL,n,m	1.1-3
DEV(d)	1.3-4
DIV	1.3-2
Einfacher Ausdruck	1.2-19
END	1.1-6
EOF(f)	1.3-3
EOLN(f)	1.3-3
EXP(a)	1.3-3
Faktor	1.2-21
FILE	1.2-5
FOR/TO/DOWNT0/DO	1.2-17
Formale Parameter	1.2-13
Formale Referenzparameter	1.2-22E
FUNCTION	1.2-11
Funktionsaufrufe	1.2-22E
Funktionsparameter	1.2-13
Funktionsvereinbarung	1.2-3
Hochfaktor	1.2-20
IF/THEN/ELSE	1.2-15
IN	1.2-9
IN	1.3-2
INTEGER	1.3-1
Konstantenvereinbarung	1.2-3
LIST,n,m	1.1-3
LN(a)	1.3-3
LOAD,u,f	1.1-2
Mengenvariable	1.2-8
MOD	1.3-2
NOT	1.3-2
ODD(a)	1.3-3
OF	1.2-16

Benutzeranleitung PASCAL Teil 1/3

Operatoren	1.3-2
OR	1.2-9
OR	1.3-2
ORD(a)	1.3-4
POWER	1.2-20
POWER	1.3-2
PRED(a)	1.3-4
PROCEDURE	1.2-11
Programm	1.2-2
Programmname	1.2-2
Prozeduraufrufe	1.2-13
Prozedurenvereinbarung	1.2-3
READ,z,d	1.1-2
READ,z,u,f	1.1-2
REAL	1.3-1
RECORD	1.2-5
RECORD	1.2-8
RECORD	1.3-1
Referenzparameter	1.2-13
REPEAT/UNTIL	1.2-18
ROUND(a)	1.3-4
RUN,d	1.1-5
SAVE,u,f	1.1-4
SCR	1.1-3
SET	1.2-5
SET	1.3-1
SIN(a)	1.3-3
SKALAR	1.3-1
Skalare Typen	1.2-6
SQR(a)	1.3-3
SQRT(a)	1.3-3
Standard-Typen	1.3-1
Standardfunktionen	1.3-3
Strukturierte Anweisungen	1.2-14
Subrange	1.2-6
Subrange	1.3-1
SUCC(a)	1.3-4
Term	1.2-20
TRUNC(a)	1.3-4
TYPE	1.2-5
TYPE	1.2-6
Typenvereinbarung	1.2-3
Umwandlungsfunktionen	1.3-4
Valueparameter	1.2-13
VAR	1.2-11
VAR	1.2-7
Variablenvereinbarung	1.2-3
Verbundanweisung	1.2-14
Vereinbarungen	1.2-2
Verzweigungsanweisungen	1.2-16
WHILE/DO	1.2-18
Ziffer	1.2-3
Zuweisungen	1.2-12

d/ietz621

PASCAL-E

Benutzeranleitung Teil 2/3
Bedienung

Benutzeranleitung PASCAL Teil 2/3

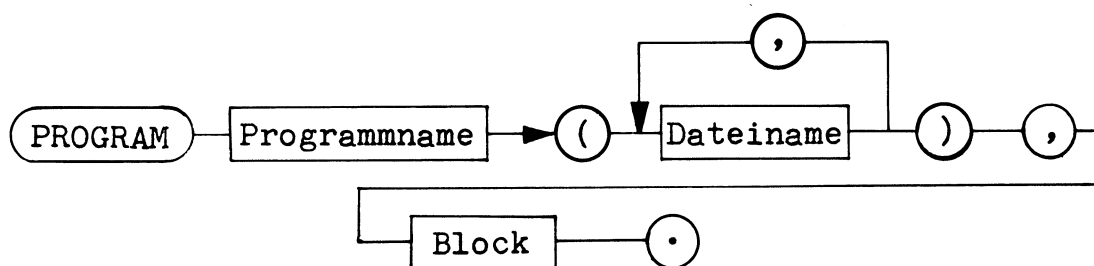
Inhaltsverzeichnis

Bedienung

- 2.0 Allgemeines
- 2.1 Kommandosprache
 - 2.1.1 Eingabe eines Beispielprogramms
 - Änderung des Quellprogramms
 - Ablage des Quellprogramms
 - Überprüfung des Quellprogramms
 - Änderung und Überprüfung
 - Starten des Programms
 - 2.1.2 Besprechung des Beispielprogramms
- 2.2 Sprachelemente
 - 2.2.1 Alphabetische Liste
 - 2.2.2 Variablenvereinbarung VAR
 - 2.2.3 Eingabeanweisung READ
 - 2.2.4 Ausgabeanweisung WRITE
 - 2.2.5 Ergibtanweisung
 - 2.2.6 Programmbeispiel DIVISION
 - 2.2.7 Wiederholungsanweisung REPEAT/UNTIL
- 2.3 Sachwörterverzeichnis

2.0 Allgemeines

Ein PASCAL-Programm hat folgenden grundsätzlichen Aufbau und läßt sich durch ein Syntax-Diagramm beschreiben.



Alle Worte bzw. Zeichen in den Kreisen und Ovalen sind in der angegebenen Form endgültig. Die Angaben in den Rechtecken müssen noch genauer definiert werden.

So z.B. 'Programmname', 'Dateiname' und 'Block'. Alle Namen müssen mit einem Buchstaben beginnen, danach können Ziffern oder Buchstaben in beliebiger Anzahl angefügt werden.

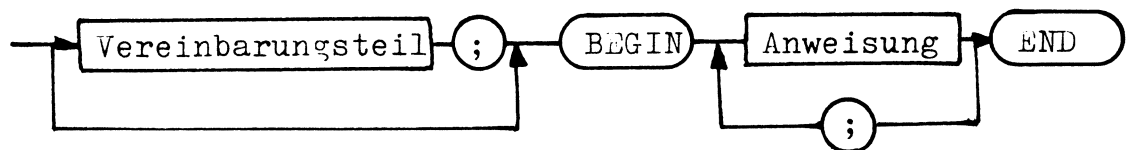
Namen sind dann unterschiedlich, wenn sie sich in den ersten 8 Zeichen unterscheiden. Als Dateinamen müssen mindestens die beiden Standard-Dateien für Ein-/Ausgabeperipherie (INPUT,OUTPUT) angegeben werden.

Ein Block besteht aus einem Vereinbarungsteil und einem Ausführungsteil. Im Vereinbarungsteil müssen alle Objekte (Konstante, Variable, Funktionen und Prozeduren) dem Compiler bekanntgegeben werden.

Benutzeranleitung PASCAL Teil 2/3

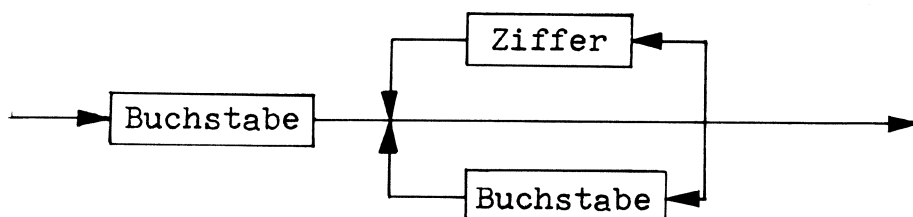
Der Ausführungsteil beginnt mit dem Schlüsselwort 'BEGIN' und enthält Anweisungen, die zur Laufzeit des Programms ausgeführt werden. Der Ausführungsteil endet mit dem Schlüsselwort 'END', das gesamte Programm mit einem '.' .

'Block'



Aus dem Syntax-Diagramm erkennt man, daß ein Block auch ohne Vereinbarungsteil sein kann. Im Ausführungsteil können dann nur konstante Objekte verwendet werden.

Die Begriffe Programmname und Dateiname basieren beide auf dem Begriff Name. Der Begriff Name läßt sich auch durch ein Syntax-Diagramm darstellen.



Benutzeranleitung PASCAL Teil 2/3

2.1 Kommandosprache

Man kann anhand eines einfachen Beispiels den grundsätzlichen Aufbau und die Eingabe eines PASCAL-Programms erkennen.

Es sollen zwei Ganzzahlen addiert werden. Die Eingabe der Zahlen erfolgt von der Konsole. Die Ausgabe der Summe soll auch auf der Konsole erfolgen.

Der Datenbereich besteht somit aus den beiden Summanden (A, B) und der Summe (SUM).

Aufgabe:PASCAL:

Schnittstelle festlegen

Ø PROGRAM SUMME (INPUT,OUTPUT);

Objekte vereinbaren

1 VAR A,B,SUM : INTEGER;

Anfang des Ausführungsteils

2 BEGIN

Eingabe von A

3 READ (A) ;

Eingabe von B

4 READ (B) ;

Summe bilden

5 SUM := A + B;

Ausgabe der Summe

6 WRITE (SUM) ;

Ende des Programms

7 END.

Die Zeilennummern (Ø bis 7) werden vom Compiler vergeben. Man kann anhand der Zeilennummern Anweisungen ändern oder löschen.

Benutzeranleitung PASCAL Teil 2/3

2.1 Kommandosprache

Die Eingabe des Beispiels erfolgt im Dialog mit dem Compiler. Die unterstrichenen Angaben sind Antworten des Compilers bzw. des Betriebssystems.

DIAL

Kommandomode des Betriebssystems

* [[RUN,] [unit,]] PASCAL <CR>Start des Compilers

PASCAL - E (Rel. 4.0)

Meldung des Compilers

* SCR <CR>

Programmspeicher löschen

PASCAL

* READ <CR>

Umschalten auf Eingabe

```
Ø PROGRAM SUMME (INPUT,OUTPUT) ; <CR>
1 VAR A B,SUM : INTEGER ; <CR>
2 BEGIN <CR>
3 READ ( A ) ; <CR>
4 READ ( B ) ; <CR>
5 SUM := A + B ; <CR>
6 WRITE (SUM) <CR>
7 END. <CR>
8 ECTRL <CR>
```

Rücksprung in den Kommandomode des Compilers.

PASCAL

*
—

Benutzeranleitung PASCAL Teil 2/3

2.1.1 Eingabe eines Beispiel-Programms

```
DIAL
*RUN,PASCAL
PASCAL-E (REL 4.00 )
```

Start des Compilers

```
PASCAL
*SCR
```

Programmspeicher
löschen

```
PASCAL
*READ
```

Umschalten auf Eingabe

```
Ø PROGRAM SUMME(INPUT,OUTPUT);
1  VAR A,B,SUM:INTEGER;
2  BEGIN
3  READ(A);
4  READ(B);
5  SUM:=A+B;
6  WRITE(SUM)
7  END.
8
```

Mit E^{CTRL} die Eingabe
beenden

```
PASCAL
*LIST
Ø PROGRAM SUMME(INPUT,OUTPUT);
1  VAR A,B,SUM:INTEGER;
2  BEGIN
3  READ(A);
4  READ(B);
5  SUM:=A+B;
6  WRITE(SUM)
7  END.
```

Gesamtprogramm listen

```
PASCAL
*LIST,2,5
2  BEGIN
3  READ(A);
4  READ(B);
5  SUM:=A+B;
```

Teilprogramm listen

Benutzeranleitung PASCAL Teil 2/3

- Änderung des Quellprogramms

PASCAL
*DEL,3

Zeile 3 löschen

```
PASCAL
*LIST
Ø PROGRAM SUMME(INPUT,OUTPUT);
1  VAR A,B,SUM:INTEGER;
2  BEGIN
3  READ(B);
4  SUM:=A+B;
5  WRITE(SUM)
6  END.
```

Gesamtprogramm listen

PASCAL
*READ,2

```
3  READ(A);
4
```

Hinter Zeile 2 ein-
fügen

Durch E^{CTRL} das Einfügen
beenden

```
PASCAL
*LIST
Ø PROGRAM SUMME(INPUT,OUTPUT);
1  VAR A,B,SUM:INTEGER;
2  BEGIN
3  READ(A);
4  READ(B);
5  SUM:=A+B;
6  WRITE(SUM)
7  END.
```

Gesamtprogramm listen

PASCAL
*ALT,4

```
4  READ(X);
5
```

Zeile 4 ändern

(Nach jeder Änderung kann
sofort eingefügt werden.
Will man das nicht, muß
man mit E-CTRL die Ein-
gabe beenden.)

```
PASCAL
*LIST
Ø PROGRAM SUMME(INPUT,OUTPUT);
1  VAR A,B,SUM:INTEGER;
2  BEGIN
3  READ(A);
4  READ(X);
5  SUM:=A+B;
6  WRITE(SUM)
7  END.
```


Benutzeranleitung PASCAL Teil 2/3

- Ablage des Quellprogramms

PASCAL
*ALT,4

Zeile 4 ändern

4 READ(B);
5

PASCAL
*SAVE,0,PENR01

Ablage des Programms
auf Laufwerk 0,
unter dem Namen PENR01

PASCAL
*LIST
0 PROGRAM SUMME(INPUT,OUTPUT);
1 VAR A,B,SUM:INTEGER;
2 BEGIN
3 READ(A);
4 READ(B);
5 SUM:=A+B;
6 WRITE(SUM)
7 END.

PASCAL
*SCR

Löschen des Programm-
speichers

PASCAL
*END

Rückkehr zum Betrieb-
systems

DIAL
*PASCAL
PASCAL-E (REL 3.04)

Start des PASCAL-E
Compilers

PASCAL
*LOAD,0,PENR01

Laden des Programm-
speichers mit dem
Quellprogramm PENR01
von Laufwerk 0.

PASCAL
*LIST
0 PROGRAM SUMME(INPUT,OUTPUT);
1 VAR A,B,SUM:INTEGER;
2 BEGIN
3 READ(A);
4 READ(B);
5 SUM:=A+B;
6 WRITE(SUM)
7 END.

Benutzeranleitung PASCAL Teil 2/3

- Überprüfung des Quellprogramms

PASCAL
*LIST

0	PROGRAM SUMME(INPUT,OUTPUT);	Gesamtprogramm
1	VAR A,B,SUM:INTEGER;	listen
2	BEGIN	
3	READ(A);	
4	READ(B);	
5	SUM:=A+B;	
6	WRITE(SUM)	
7	END.	

PASCAL
*COMP

0	PROGRAM SUMME(INPUT,OUTPUT);	Programm auf formale
1	VAR A,B,SUM:INTEGER;	Richtigkeit überprüfen
2	BEGIN	
3	READ(A);	
4	READ(B);	
5	SUM:=A+B;	
6	WRITE(SUM)	
7	END.	

COMPILATION TERMINATED, NO ERRORS FOUND

Um zu demonstrieren, wie der PASCAL-E Dialog-Compiler Fehler meldet, sind in den folgenden Beispielen absichtlich Fehler eingegeben worden.

Der Compiler liefert eine sehr detaillierte Fehlerbeschreibung. Es wird nicht nur die fehlerhafte Zeile, sondern auch die fehlerhafte Stelle in der Zeile gekennzeichnet.

Zusätzlich wird eine verbale Beschreibung der Fehler ausgegeben.


Benutzeranleitung PASCAL Teil 2/3

- Änderung und Überprüfung des Quellprogramms

PASCAL
*ALT,0

Zeile 0 ändern

```
0 PROGRAM SUMME(INPUT,OUTPUT)
1
```

 Semikolon fehltPASCAL
*COMP

Programm überprüfen


```
0 PROGRAM SUMME(INPUT,OUTPUT)
1 VAR A,B,SUM:INTEGER;
***** 9014 ~
2 BEGIN
3 READ(A);
4 READ(B);
5 SUM:=A+B;
6 WRITE(SUM)
7 END.
```

EXPLANATION OF ERRORS

9014 ';' ODER : ODER (' ERWARTET

PASCAL
*ALT,1

```
1 VAR A,SUM:INTEGER
2
```

Variable B ist nicht
mehr definiertPASCAL
*COMP

```
0 PROGRAM SUMME(INPUT,OUTPUT)
1 VAR A,SUM:INTEGER
***** 9014 ~
2 BEGIN
***** 9014 ~
3 READ(A);
4 READ(B);
***** 9000 ~
5 SUM:=A+B;
***** 9000 ~
6 WRITE(SUM)
7 END.
```

EXPLANATION OF ERRORS

9000 OBJEKT NICHT DEFINIERT

9014 ';' ODER : ODER (' ERWARTET

Benutzeranleitung PASCAL Teil 2/3

- Starten des Programms im Dialog

```
PASCAL
*LIST
0  PROGRAM SUMME(INPUT,OUTPUT);
1  VAR A,B,SUM:INTEGER;
2  BEGIN
3  READ(A);
4  READ(B);
5  SUM:=A+B;
6  WRITE(SUM)
7  END.
```

```
PASCAL
*RUN
10 10
      20
```

Die Eingabe der beiden Werte wird
durch ' ' getrennt.
Abschluß der Eingabe ist
die Taste <cr> .

```
PASCAL
*RUN
1 2345
      2346
```

```
PASCAL
*RUN
3 3
      6
```

```
PASCAL
*1001 222
ERR: 97
```

—————→ Kommando nicht bekannt

```
PASCAL
*RUN
1001 222
      1223
```

```
PASCAL
*100 -333
ERR: 97
```

```
PASCAL
*RUN
100 -333
      -233
```

```
PASCAL
*RUN
20000 -19999
      1
```

```
PASCAL
*
```

Benutzeranleitung PASCAL Teil 2/3

2.1.2 Besprechung des Beispielprogramms

In der ersten Zeile wurde neben dem Programmnamen auch die Datenschnittstelle (INPUT, OUTPUT) festgelegt. Durch das Schlüsselwort VAR eingeleitet, werden die veränderbaren Objekte (Variable) bekanntgegeben. Drei Programmobjekte SUM, A und B werden als Standardtypen Integer festgelegt. Der Wertebereich einer Integervariablen liegt zwischen -2^{31} und $+2^{31}-1$. Daneben gibt es noch mehr Standardtypen und frei definierbare Typen. Der danach folgende Ausführungsteil wird als Verbundanweisung bezeichnet. Eine Verbundanweisung beginnt mit dem Schlüsselwort BEGIN und endet mit dem Schlüsselwort END. Die Anweisungen zwischen BEGIN und END müssen voneinander durch Semikolon (;) getrennt werden. Die Verbundanweisung besteht aus zwei Standardprozeduraufrufen "READ (A); READ (B)", einer Ergibtanweisung "SUM:= A + B" und einem weiteren Standardprozeduraufruf "WRITE (SUM)". Standardprozeduren müssen nicht vereinbart (deklariert) werden.

Benutzeranleitung PASCAL Teil 2/3

2.2.1 Alphabetische Liste

Ergibtanweisung ':='	2.2 - 9
INTEGER	2.2 - 3
Programmbeispiel DIVISION	2.2 - 10
READ	2.2 - 4
REAL	2.2 - 2
REPEAT/UNTIL	2.2 - 12
VAR	2.2 - 2
WRITE	2.2 - 5
WRITELN	2.2 - 5

Benutzeranleitung PASCAL Teil 2/3

2.2.2 Variablenvereinbarungen

Die Definition von Variablen ist Bestandteil des Vereinbarungsteils.

Beispiel:

```
VAR □ QUOTIENT : REAL
```

Hinter dem Schlüsselwort VAR folgt, durch ein Leerzeichen (**Blank**) getrennt, der Name der zu definierenden Variablen. Im Anschluß an das Trennzeichen ":" muß der Typ der Variablen, hier REAL, angegeben werden.

Sollen mehrere Variablen vom gleichen Typ in einer Anweisung definiert werden, so geschieht dies durch Aufzählen der entsprechenden Variablennamen, durch Kommata getrennt.

Beispiel:

```
VAR □ QUOTIENT, DIVIDEND, DIVISOR: REAL
```

Mit dieser Anweisung werden die drei genannten Variablen mit dem Typ REAL definiert.

REAL bedeutet reelle Zahlen mit Exponentialdarstellung. Intern werden Variable vom Typ REAL in 8 Byte dargestellt.

Wertbereich: von -2^{55} bis $+2^{55}$

Benutzeranleitung PASCAL Teil 2/3

Sollen in einer Anweisung mehrere Variablen verschiedener Typen definiert werden, ist folgende Schreibweise zu beachten:

Beispiel:

```
VAR  □ QUOTIENT:REAL;  
      DIVIDEND,DIVISOR:INTEGER
```

Mit dieser Anweisung werden die Variablen QUOTIENT, vom Typ REAL, DIVIDEND und DIVISOR, beide vom Typ INTEGER, definiert.

INTEGER-Variablen können die Werte der ganzen Zahlen im Bereich von -2^{31} bis $+2^{31}$ annehmen.

2.2.3 Eingabeanweisung READ

Die Anweisung READ ermöglicht das Einlesen von Werten in eine Variable. Der Name der Variablen wird hinter dem Wort READ in Klammern genannt.

Beispiel:

```
READ(EINGABE)
```

Mit dieser Anweisung besteht die Möglichkeit, für die Variable mit dem Namen EINGABE einen Wert einzulesen. Man beachte, daß der Wert, der eingegeben wird, und die Variable typidentisch sein müssen.

Sollen mehrere Werte für verschiedene Variablen eingegeben werden, so kann dies durch Aufzählen der Variablennamen, durch Kommata getrennt, erfolgen.

Beispiel:

```
READ (EINGABE,ANZAHL,NUMMER)
```

Die Wertzuweisungen erfolgen in der angegebenen Reihenfolge. Man beachte, daß die Eingaben für die einzelnen Variablen durch eine Leertaste (Blank) voneinander getrennt werden müssen.

Benutzeranleitung PASCAL Teil 2/3

2.2.4 Ausgabeanweisung WRITE

Mit der Anweisung WRITE wird der Wert eines Ausdrucks ausgegeben. Der Ausdruck wird hinter dem Wort WRITE in Klammern genannt. Sehr häufig ist der Ausdruck ein Variablenname.

Beispiel:

WRITE (ERGEBNIS)

Diese Anweisung gibt den Wert der Variablen ERGEBNIS aus.

Ebenso wie bei der READ-Anweisung können auch hier Ausdrücke, durch Kommata getrennt, aufgezählt werden.

Beispiel:

WRITE (SUMME,PRODUKT,WURZEL)

Die Werte der genannten Variablen werden in der angegebenen Reihenfolge ausgegeben.

Soll die Ausgabe mit einer Zeilenschaltung abgeschlossen werden, so ist die Anweisung

WRITELN (Ausdruck)

zu verwenden.

WRITELN bedeutet WRITE-LINE.

Benutzeranleitung PASCAL Teil 2/3

Soll die Ausgabe auf ein bestimmtes Gerät erfolgen, so geschieht dies durch Verwenden der Systemfunktion DEV (d) als Argument hinter der WRITE-Anweisung.

Beispiel:

```
WRITE (DEV(3),SUMME, PRODUCT)
```

Die Ausgabe der Werte der Variablen SUMME und PRODUKT erfolgt auf dem Gerät (Device) mit der logischen Nummer 3. Standardmäßig ist das Gerät Nr. 3 ein Drucker.

Man beachte, daß die Auswahl eines Gerätes für alle nachfolgenden WRITE- und READ-Anweisungen gilt. Das Anwählen eines Gerätes ist nur mit einer WRITE-Anweisung möglich.

Um eine bessere Übersicht bei der Ausgabe zu erhalten, können neben den Werten der Ausdrücke auch Texte ausgegeben werden. Der Text ist eine beliebige Zeichenfolge, die in Apostrophe eingeschlossen, als Argument in der WRITE-Anweisung steht.

Beispiel:

```
WRITE ('SALDO',SUMME,'NETTO',PREIS)
```

Es erfolgen die Ausgaben:

- SALDO,
- Wert der Variablen SUMME,
- NETTO,
- Wert der Variablen PREIS

Als Argument in einer WRITE-Anweisung ist eine Format-Angabe möglich. Ein solches Argument besteht dann aus drei Ausdrücken, die durch das Zeichen " : " voneinander getrennt sind.

Beispiel:

```
WRITE (SUMME:10:2)
```

Die Variable SUMME sei vom Typ REAL und habe den Wert 456.319.

Somit erfolgt die Ausgabe

```
□□□□456.31
```

Benutzeranleitung PASCAL Teil 2/3

Der erste Ausdruck innerhalb des Arguments bestimmt den auszugebenden Wert. Die beiden folgenden Ausdrücke bestimmen das Format. Die Formatdefinition geschieht durch Angabe der insgesamt auszugebenden Stellen und durch Angabe der auszugebenden Nachkommastellen.

Man beachte, daß der Dezimalpunkt und das Vorzeichen bei der Angabe der Gesamtstellenzahl berücksichtigt werden müssen.

Benutzeranleitung PASCAL Teil 2/3

2.2.5 Ergibt-Anweisung

Mit einer Ergibt-Anweisung wird einer Variablen der Wert eines Ausdrucks zugewiesen. Im einfachsten Fall ist dieser Ausdruck eine Konstante.

Beispiel:

ANZAHL : = 50

Der Variablen ANZAHL wird der Wert 50 zugewiesen.

Beispiel:

QUOTIENT : = DIVIDEND/DIVISOR

Der Variablen QUOTIENT wird der Wert des Ausdrucks zugewiesen. Der Ausdruck ist in diesem Fall der Quotient aus dem Wert der Variablen DIVIDEND und dem Wert der Variablen DIVISOR.

Benutzeranleitung PASCAL Teil 2/3

2.2.6 Programmbeispiel DIVISION

```
PASCAL
*COMP
0      PROGRAM DIVISION(INPUT,OUTPUT);
1      (*VEREINBARUNGSTEIL*)
2      VAR QUOTIENT,DIVIDEND,DIVISOR:REAL;
3      (*AUSFUEHRUNGSTEIL*)
4      BEGIN
5      READ(DIVIDEND,DIVISOR);
6      QUOTIENT:=DIVIDEND/DIVISOR;
7      WRITE(QUOTIENT)
8      END
9      .
```

COMPILATION TERMINATED; NO ERRORS FOUND

```
PASCAL
*
```

```
RUN
2345 543
.4318600368E 01
```

```
PASCAL
*RUN
12 4
.30000000000E 01
```

```
PASCAL
*
```

Die Eingaben für die beiden Variablen DIVIDEND und DIVISOR müssen durch eine Leertaste (**Blank**) voneinander getrennt werden. Zum Abschluß der Eingaben muß die Taste CR betätigt werden.

Da die Variable QUOTIENT vom Typ REAL ist, erfolgt die Ausgabe des Wertes dieser Variablen in Gleitkommadarstellung mit dem Exponenten E zur Basis 10.

Durch Ändern der Programmzeile 7 in

```
WRITE('ERGEBNIS := ',QUOTIENT:10:2);
```

erhält man eine formatierte Ausgabe mit einer Textausgabe.

Benutzeranleitung PASCAL Teil 2/3

Die beiden Eingabevariablen DIVIDEND und DIVISOR sind vom Typ REAL. Somit besteht die Möglichkeit, auch Exponenten und gebrochene Zahlen einzugeben.

In der herkömmlichen Schreibweise werden der ganze und gebrochene Teil einer Zahl durch ein Komma getrennt. Anstelle dieses Kommas muß hier ein Punkt stehen.

```
PASCAL
*RUN
1E5 .1E5
ERGEBNIS :=      10.00
```

```
PASCAL
*RUN
-56 5.7
ERGEBNIS :=      -9.82
```

```
PASCAL
*
```


2.2.7 Wiederholungsanweisung REPEAT UNTIL

Soll ein bestimmtes Programmstück wiederholt werden, so ist dies mit der Anweisung REPEAT/UNTIL möglich. Alle Anweisungen, die zwischen den Schlüsselwörtern REPEAT und UNTIL stehen, werden **solange wiederholt**, bis die Bedingung, die hinter dem Schlüsselwort UNTIL steht, zutrifft. Die Abbruchbedingung ist ein Ausdruck, dessen Wert die Zustände 'wahr' (TRUE) oder 'falsch' (FALSE) annehmen kann. Ein solcher Ausdruck heißt logischer oder vergleichender Ausdruck.

Beispiel:

```
I := 0;  
REPEAT  
    I := I + 1;  
    WRITELN (I)  
UNTIL I = 10
```

Der Variablen I werden nacheinander die Werte 1 bis 10 zugewiesen und jeder Wert wird in einer Zeile ausgegeben.

Der Ausdruck hinter dem Schlüsselwort UNTIL wird TRUE, wenn die Variable I den Wert 10 hat; damit ist die Abbruchbedingung erfüllt.

Stehen mehrere Anweisungen zwischen den Schlüsselwörtern REPEAT und UNTIL, so zählen diese Anweisungen im Programm als eine einzige Anweisung; sie brauchen also nicht in BEGIN und END eingefaßt zu werden.

Benutzeranleitung PASCAL Teil 2/3

2.3 Sachwörterverzeichnis

Ablage Quellprogramm	2.1-5
Änderung Quellprogramm	2.1-4
Anweisung	2.0-2E
Ausführungsteil	2.0-1
BEGIN/END	2.0-2E
Besprechung Beispielprogramm	2.1-9E
BLOCK	2.0-1
Dateiname	2.0-1
DEV(d)	2.2-6
Eingabe Programmbeispiel	2.1-3
Ergibt-Anweisung	2.2-9
FALSE	2.2-12
Format-Angabe	2.2-7
INTEGER	2.2-3
Kommandosprache	2.1-1
Kommandosprache	2.1-2
Name	2.0-2E
PROGRAM	2.0-1
Programmbeispiel DIVISION	2.2-10
Programmname	2.0-1
READ	2.2-4
REAL	2.2-2
REPEAT/UNTIL	2.2-12
Starten Programm	2.1-8
Syntax-Diagramm	2.0-2E
TRUE	2.2-12
Überprüfung Quellprogramm	2.1-6
VAR	2.2-2
Vereinbarungsteil	2.0-1
Vereinbarungsteil	2.0-2E
WRITE	2.2-5
WRITELN	2.2-5

dietz621

PASCAL-E

Benutzeranleitung Teil 3/3
Beispiele

Benutzeranleitung PASCAL Teil 3/3

InhaltsverzeichnisBeispielsammlung

<u>Aufgabe</u>	<u>Seite</u>
Mittelwert	3.-1
PASCAL-Dreieck	3.-2
Faktorielle	3.-2
Teiler	3.-3
Integration	3.-3
Ewiger Kalender	3.-4
Bestimmung aller Zwei- bzw. vierziffrigen Zahlen	3.-5
Lissajousche Figuren	3.-6
Sinus Kurve	3.-6
Gauss	3.-7
Umwandlung	3.-8
Dezimal-Dual	
Holz-Spiel	3.-9
Sortieren	3.-10

Benutzeranleitung PASCAL Teil 3/3

```
PROGRAM MITTELWERT(INPUT,OUTPUT);
(*-----*)

VAR WERT,SUMME,ANZAHL : INTEGER;
    ERGEBNIS,REST      : INTEGER;

BEGIN  SUMME:=0;ANZAHL:=0;
WRITELN('EINGABE VON WERTEN(ENDE BEI 0-EINGABE)');
READ(WERT);

REPEAT
  WRITELN(WERT);
  SUMME:= SUMME+WERT;
  ANZAHL:=ANZAHL+1;
  READ(WERT)
UNTIL WERT=0;

ERGEBNIS:=SUMME DIV ANZAHL;REST:=SUMME MOD ANZAHL;
WRITELN('MITTELWERT VON ',SUMME,' : ',ANZAHL,' IST ',ERGEBNIS,
' REST IST ',REST);
END.

COMPILATION TERMINATED, NO ERRORS FOUND
```

Benutzeranleitung PASCAL Teil 3/3

```

0      PROGRAM PASCAL(OUTPUT);
1      (* AUSGABE DES PASCALSCHEN DREIECKES *)
2
3
4      CONST N=13;
5      VAR I,J:INTEGER;
6          A: ARRAY[1..N] OF INTEGER;
7
8
9      BEGIN
10     FOR I:=1 TO N DO
11         BEGIN ACJ:=1;
12             FOR J:= I-1 DOWNT0 2 DO
13                 ACJ:=ACJ-1] +ACJ];
14                 WRITE(' ':2*(N-I)+1);
15                 FOR J:= 1 TO I DO WRITE(ACJ:4);
16             WRITELN
17             END
18     END.
    
```

COMPILATION TERMINATED, NO ERRORS FOUND

```

0      PROGRAM FAKTORIELLE(OUTPUT);
1      (* BERECHNUNG FAKULTAET ITERATIV UND REKURSIV *)
2
3      CONST MAX=8;
4
5      VAR I,Y:INTEGER;
6
7
8      FUNCTION FAKT(X:INTEGER):INTEGER;
9      BEGIN IF X<>1 THEN FAKT:=FAKT(X-1)*X ELSE FAKT:=1 END;
10
11
12     BEGIN Y:=1;
13     WRITELN('BERECHNUNG FAKULTAET ITERATIV');
14     FOR I:=1 TO MAX DO
15         BEGIN Y:=Y*I;
16             WRITELN(I:10,Y:15)
17         END;
18     WRITELN('BERECHNUNG FAKULTAET REKURSIV');
19     FOR I:=1 TO MAX DO WRITELN(I:10,FAKT(I):15)
20     END.
    
```

COMPILATION TERMINATED, NO ERRORS FOUND

Benutzeranleitung PASCAL Teil 3/3

```

0      PROGRAM TEILER(INPUT,OUTPUT);
1      CONST N=6;
2      VAR I,G:INTEGER;
3          X: ARRAY[1:N] OF INTEGER;
4      FUNCTION GGT(A,B:INTEGER):INTEGER;
5          VAR R:INTEGER;
6      BEGIN
7          WHILE B(>) 0 DO BEGIN R:=A MOD B;A:=B;B:=R END;
8          GGT:=A
9          END;
10     BEGIN
11     WRITELN('EUKLIDALGORITHMUS= DIVISION VON 6 ZAHLEN ');
12     WRITELN('DURCH IHREN GROESSTEN GEMEINSAMEN TEILER ');
13     WRITELN('EINGABE DER ZAHLEN ');
14     FOR I:=1 TO N DO READ(X[I]);
15     G:= GGT(X[1],X[2]);
16     FOR I:=3 TO N DO G:=GGT(X[I],G);
17     FOR I:=1 TO N DO WRITELN(X[I]:10,X[I] DIV G)
18     END.

```

COMPILATION TERMINATED, NO ERRORS FOUND

```

0      PROGRAM INTEGRATION(OUTPUT);
1      FUNCTION INTEGRAL(A,B:REAL;FUNCTION F:REAL):REAL;
2          CONST N=100;
3          VAR S,H:REAL;
4              I,K:INTEGER;
5      BEGIN
6          H:=(B-A)/N; K:=2;
7          S:=F(A)+F(B);
8          FOR I:=1 TO N-1 DO
9              BEGIN K:=6-K;
10             S:=S+K*F(A+I*H)
11             END;
12          INTEGRAL:= H/3.0*S;
13      END;
14
15     FUNCTION Y(X:REAL):REAL;
16     BEGIN Y:=1/(1+X) END;
17
18     BEGIN
19     WRITELN(INTEGRAL(0.,1.0,Y))
20     END.

```

COMPILATION TERMINATED, NO ERRORS FOUND

Benutzeranleitung PASCAL Teil 3/3

```

0      PROGRAM EWIGKAL(INPUT,OUTPUT);
1      VAR K,M,C,D,X:INTEGER;
2          C1:REAL;
3      BEGIN(* EWIGER KALENDER *)
4      REPEAT
5          WRITELN('WELCHES DATUM');
6          READ(K,M,C);
7          WRITE('AM',K:2,'.',M:2,'.',C:4,' IST ');
8          C1:=C/100;D:=TRUNC(0.4+100*(C1-TRUNC(C1)));C:=TRUNC(C1);
9          M:=M-2;
10         IF M<=0 THEN
11             BEGIN
12                 M:=M+12;D:=D-1;
13                 IF D<=0 THEN
14                     BEGIN
15                         D:=99;C:=C-1;
16                     END;
17             END;
18         X:=(26*M-2)DIV 10+K+D+C DIV 4+D DIV 4-2*C;
19         WHILE X>=6 DO X:=X-7;
20         CASE X OF
21             0:WRITELN('SONNTAG');
22             1:WRITELN('MONTAG');
23             2:WRITELN('DIENSTAG');
24             3:WRITELN('MITTWOCH');
25             4:WRITELN('DONNERSTAG');
26             5:WRITELN('FREITAG');
27             OTHERS:WRITELN('SAMSTAG');
28         END;
29     UNTIL FALSE;
30     END.

```

COMPILATION TERMINATED, NO ERRORS FOUND

Benutzeranleitung PASCAL Teil 3/3

```

PROGRAM TUENS4 (OUTPUT);
  (* BESTIMMUNG ALLER ZWEI- BZW VIERZIFFRIGEN
     ZAHLEN MIT: A*B = A HOCH B
     BZW: A*B*C*D = (A HOCH B)*(C HOCH D) *)
  TYPE FELD = ARRAY[1..4] OF INTEGER;
  VAR N,U,V: INTEGER;
      A: FELD;
  PROCEDURE ZERLEGE (Z:INTEGER; VAR A:FELD);
    VAR S,I: INTEGER;
  BEGIN S:=1;
        REPEAT S:=S*10
          UNTIL S>Z;
        I:=1;
        REPEAT S:=S DIV 10;
              AC1:=Z DIV S;
              Z:=Z MOD S;
              I:=I+1
          UNTIL S=1
        END;
  FUNCTION POTENZ (A,B:INTEGER):INTEGER;
    VAR P,I: INTEGER;
  BEGIN P:=1;
        FOR I:=1 TO B DO P:=P*A;
        POTENZ:=P
    END;

  BEGIN N:=10;
    WRITELN (' FUER FOLGENDE ZWEI- BZW VIERZIFFRIGEN');
    WRITE (' ZAHLEN GILT:');
    WRITELN (' ':11, 'A*B = A HOCH B');
    WRITELN (' BZW: A*B*C*D = (A HOCH B)*(C HOCH D)');
    WRITELN;
    WHILE ((N)>10) AND (N<=99) OR ((N)>1010) AND
      (N<=9999) DO
      BEGIN ZERLEGE(N,A);
        IF (N)>10) AND (N<=99)
          THEN BEGIN U:=POTENZ(AC1,AC2);
                   V:=AC1*10+AC2
                END
          ELSE BEGIN U:=POTENZ(AC1,AC2)*POTENZ(AC3,AC4)
                END;
        IF U=V
          THEN IF N<=99
              THEN BEGIN WRITELN (' ':7, N, ' = ', AC1);
                       WRITE (' HOCH ', AC2)
                     END
                ELSE BEGIN WRITELN (' ':5, N, ' = (', AC1);
                       WRITE (' HOCH ', AC2, ') * (', AC3);
                       WRITE (' HOCH ', AC4, ')') END
          ELSE N:=N+1
        END
      END.

  COMPILATION TERMINATED, NO ERRORS FOUND
  
```

Benutzeranleitung PASCAL Teil 3/3

```

0      PROGRAM LISSAC(INPUT,OUTPUT);
1      CONST PI=3.1415296;
2          STEP=0.025;
3      VAR A:ARRAY[1:25,1:50] OF BOOLEAN;
4          I1,I2:INTEGER;
5          X2,X1,X,PH,FQ:REAL;
6      BEGIN
7          WRITELN('LISSAJOUSCHE FIGUREN');
8          FOR I1:=1 TO 25 DO
9              FOR I2:=1 TO 50 DO
10                 ACI1,I2]:=FALSE;
11          WRITELN('EINGABE PHASENVERSCHIEBUNG IN GRAD');
12          READ(PH);
13          WRITELN('EINGABE FREQUENZQUOTIENT');
14          READ(FQ);
15          PH:=PH/180*PI;
16          X:=0;
17          WHILE X<2*PI DO
18              BEGIN
19                  X2:=SIN(X);X1:=SIN(FQ*(X+PH));
20                  I2:=ROUND((X2+1)*24+1);I1:=ROUND((X1+1)*42+1);
21                  ACI1,I2]:=TRUE;
22                  X:=X+STEP;
23              END;
24          WRITELN;
25          FOR I1:=1 TO 25 DO
26              BEGIN
27                  FOR I2:=1 TO 50 DO
28                      IF ACI1,I2] THEN WRITE('*') ELSE WRITE(' ');
29                  WRITELN;
30              END;
31          END.

```

COMPILATION TERMINATED, NO ERRORS FOUND

```

0      PROGRAM KURVE(OUTPUT);
1      CONST PI=3.14159265;
2      VAR X,Y : REAL;
3      BEGIN X:=-3*PI;
4      REPEAT
5
6          IF X=0 THEN Y:=1
7              ELSE Y:=SIN(X)/X;
8          WRITELN(' ':ROUND(30*Y)+30,'*');
9          X:= X+PI/10;
10         UNTIL X>3*PI
11     END.

```

COMPILATION TERMINATED, NO ERRORS FOUND

Benutzeranleitung PASCAL Teil 3/3

```

0      PROGRAM GAUSS(INPUT,OUTPUT);
1      VAR I,J,K,N,L,L1:INTEGER; Q:REAL;
2          A:ARRAY[1..9,1..9] OF REAL;
3          B,X:ARRAY[1..9] OF REAL;
4      PROCEDURE AUSGABE;
5      BEGIN
6          WRITELN(' AUSGABE DER KOEFFIZIENTEN');
7      FOR L:=1 TO N DO
8          BEGIN
9              FOR L1:=1 TO N DO
10                 WRITE(ACL,L1:10:4,' ');
11                 WRITE(BC[L]:10:4); WRITELN
12             END;
13      END;
14
15      BEGIN (*EINLESEN DER KOEFFIZIENTEN UN DER RECHT SEITE*)
16      READ(N);
17      FOR I:=1 TO N DO
18          BEGIN
19              FOR J:=1 TO N DO
20                 READ(ACI,J);
21                 READ(BCI); WRITELN
22             END;
23
24      AUSGABE;
25      FOR I:=1 TO N-1 DO (*ELIMINATION VON X[I]*)
26          FOR K:=I+1 TO N DO (*AUS DER KTEN GLEICHUNG*)
27              BEGIN Q:= ACK,I]/ACI,I];
28                  FOR J:=I+1 TO N DO ACK,J]:=ACK,J]-ACI,J]*Q;
29                  BEK]:=BEK]-BCI]*Q;
30              END;
31      FOR I:=N DOWNTO 1 DO(*BERECHNEN VON X[I]*)
32          BEGIN
33              FOR J:=I+1 TO N DO BCJ]:=BCJ]-ACI,J]*X[J];
34              XCI]:=BCI]/ACI,I]
35          END;
36      WRITELN;
37      FOR I:=1 TO N DO (*AUSGABE VON XCI*)
38          WRITELN(' X',I:1,' =',XCI:24:2)
39      END.

```

COMPILATION TERMINATED, NO ERRORS FOUND

Benutzeranleitung PASCAL Teil 3/3

```
0      PROGRAM ZAHLENUMWANDLUNG(INPUT,OUTPUT);
1      (* UMWANDLUNG DEZIMAL  BINAER *)
2
3      CONST BASIS=2;
4      VAR  Z,S,SCHLUSS:INTEGER;
5      BEGIN
6
7      WRITELN('BINAERDARSTELLUNG EINER ZAHL (<=16384)');
8      REPEAT
9          REPEAT  BEGIN
10
11                      WRITELN('EINGABE (0 IST END) ');
12                      READ(Z);
13                      END
14                      UNTIL Z<=16383;
15
16      SCHLUSS:=Z;
17      WRITE(Z,' = ');
18      S:=1;
19      REPEAT
20          S:=S*BASIS
21          UNTIL  S> Z;
22
23      REPEAT
24          S:=S DIV BASIS;
25          WRITE(Z DIV S:1);
26          Z:=Z MOD S
27
28          UNTIL S=1;
29      WRITELN;
30      UNTIL SCHLUSS=0
31      END.
```

COMPILATION TERMINATED, NO ERRORS FOUND

Benutzeranleitung PASCAL Teil 3/3

```
PROGRAM NIMM(INPUT,OUTPUT);
(* NIM- SPIEL      *)

CONST TAUS=1000;
VAR N,J,MAX:INTEGER;

(* PROZEDUR EINLESEN *)
PROCEDURE EINLESEN(VAR X,Y:INTEGER);
BEGIN
  WRITELN('WIEVIEL STAEBCHEN ?');WRITELN;
  READLN;READ(X);
  IF (X<0) OR (X>Y) THEN BEGIN
    WRITELN('ANZAHL FALSCH ');
    EINLESEN(X,Y);
  END
END (* END EINLESEN *);

BEGIN
  J:=TAUS;
  EINLESEN(N,J);

  REPEAT J:=(N+3) MOD 4;
    IF J=0 THEN J:=1
      ELSE WRITELN('SIE HABEN KEINE CHANCE MEHR');
    WRITELN('ICH NEHME',J:3);N:=N-J;
    WRITELN('ES BLEIBEN ',N:3,' STAEBCHEN');
    IF N=1 THEN BEGIN
      WRITELN('SIE HABEN VERLOREN!');
      N:=0;
      END
    ELSE BEGIN
      WRITE('IHR ZUG:');WRITELN;
      IF N<3 THEN MAX:=N ELSE MAX:=3;
      EINLESEN(J,MAX); N:=N-J;
      WRITELN('ES BLEIBEN ',N:3,' STAEBCHEN')
      END
    UNTIL N<=1;
  IF N=1 THEN WRITELN('GRATULIERE, SIE HABEN GEWONNEN!');
  WRITELN
END.

COMPILATION TERMINATED, NO ERRORS FOUND
```

Benutzeranleitung PASCAL Teil 3/3

```
PROGRAM SORTIEREN (INPUT,OUTPUT);
CONST N=100;
VAR   I,J,MAX: INTEGER;
      SORTIERT: BOOLEAN;
      A:      ARRAY[1:N] OF INTEGER;

PROCEDURE
  TAUSCHE(VAR X,Y:INTEGER);
  VAR Z:INTEGER;
  BEGIN Z:=X;X:=Y;Y:=Z      END;

BEGIN  (* EINGABE *)

  WRITELN(' EINGABE VON GANZZAHLEN 1-100 ');
  WRITELN(' MAXIMALZAHL 1-100');
  READ(MAX);
  WRITELN(' ZAHLENEINGABE ');
  FOR I:=1 TO MAX DO READ(A[I]);

  (* AUSGABE DES UNSORTIERTEN FELDES *)

  FOR I:=1 TO MAX DO WRITE(A[I]:5);
  WRITELN;

  (* SORTIEREN *)

  J:=MAX;
  REPEAT  SORTIERT:= TRUE;J:=J-1;
          FOR I:=1 TO J DO
              IF A[I]>A[I+1] THEN BEGIN
                  TAUSCHE(A[I],A[I+1]);
                  SORTIERT:= FALSE
              END
          UNTIL  SORTIERT;

  (*AUSGABE DES SORTIERTEN FELDES *)
  FOR I:=1 TO MAX DO  WRITE(A[I]:5); WRITELN
  END.

  COMPILATION TERMINATED, NO ERRORS FOUND
```


Heinrich Dietz
Solinger Straße 9
4330 Mülheim-Ruhr
Tel.: (0208) 4434-1
Telex 856770

DIETZ Computer
SYSTEME