

## Einfache Ein-/Ausgabe-Prozeduren

**Mit Pascal lernt man, strukturierte Programme zu schreiben, die sich auch selbst aufrufen können (rekursiv). Das kann Basic nicht!**

Die meisten Besitzer eines Commodore 64 programmieren ihren Computer in Basic oder vielleicht noch in Maschinensprache. Inzwischen erfreut sich Pascal auch auf dem C 64 zunehmender Beliebtheit. Pascal ist eine leicht erlernbare Sprache, in der viele Probleme eleganter und einfacher zu lösen sind als in Basic. Der Benutzer muß sich aber über die Struktur seines Programms im klaren sein, bevor er mit dem Programmieren anfängt. Pascal erzieht zu einem systematischen Vorgehen und zu einem problemnahen Programmieren. Deswegen fällt es Basic-Programmierern bisweilen schwer, sich an die starre Struktur von Pascal zu gewöhnen. Anfängern erleichtert Pascal jedoch den Einstieg ins Programmieren.

### Ein Philosoph stand Pate

Pascal wurde 1971 von Niklaus Wirth in Zürich entwickelt. Die von Wirth definierte Sprache lief zunächst nur auf Großrechnern. Seinen Siegeszug auf Mikrocomputern trat Pascal an, als auf der Universität von California in San Diego (UCSD) das berühmte UCSD-Pascal entstand. Diese Version enthält einige Abweichungen von Standard-Pascal; sowohl Einschränkungen als auch Erweiterungen. Pascal wurde nach dem berühmten französischen Mathematiker und Philosophen Blaise Pascal benannt.

Inzwischen gibt es für die meisten Computer mehrere Pascal-Compiler. Auf dem Commodore 64 haben sich Oxford-Pascal und Profi-Pascal bei unseren Tests (siehe Ausgabe 4/85 und 8/85) als besonders geeignet erwiesen. Deswegen wurden alle Beispiele in dieser Serie mit diesen Versionen getestet. Auch wer eine andere Pascal-Version besitzt, wird diese Serie mit Gewinn verfolgen, da die meisten Beispiele auf allen Compilern laufen werden. Erweiterungen und Abweichungen werden besonders besprochen.

Ein wesentlicher Unterschied zwischen Basic und Pascal besteht in der Ausführung des fertigen Programms. Commodore-Basic



läuft unter einem Interpreter. Das bedeutet: während des Programmlaufs wird jede Basic-Anweisung vom Interpreter gelesen, entschlüsselt und dann ausgeführt. Dies hat den Vorteil, daß sich Basic-Programme besonders schnell entwickeln lassen und Fehler leicht ausgebessert werden können. Der Nachteil ist in den längeren Laufzeiten zu sehen. Verwendet man zum Beispiel eine FOR-NEXT-Schleife mit mehreren Befehlen, so werden diese Befehle so oft interpretiert, solange die Schleife läuft.

### Compiler kontra Interpreter

Compiler übersetzen im Gegensatz dazu den Quellcode (Programm in einer Hochsprache) in Maschinensprache. Meist wird jedoch ein Zwischenweg eingeschlagen. Der Compiler übersetzt den Quellcode in einen Zwischencode, der auch P-Code genannt wird. Dieser Zwischencode wird dann ebenfalls wieder interpretiert. Ein solches übersetztes Programm läuft natürlich viel schneller ab. Der Nachteil besteht im zusätzlichen (zeitlichen) Aufwand für die Übersetzung des Programms. Außerdem wird zum Erstellen des Programms ein Editor benötigt, der erst von der Diskette geladen werden muß. Generell kann man sagen, daß die Programmierung mit einem Compiler umständlicher, die Laufzeiten des Programms aber kürzer sind.

### Aus Quellcode wird P-Code

Der Oxford-Compiler übersetzt das Quellprogramm in P-Code, Profi-Pascal ebenfalls. Oxford-Pascal arbeitet auf zwei Operations-Ebenen. Im Resident-Modus befindet sich der Compiler zusammen mit dem Benutzer-Programm (Quellcode und übersetzter P-Code) im Arbeitsspeicher des Computers. Das Programm kann übersetzt und gleich ausgeführt werden. Im Disk-Modus wird der

Compiler aus dem Arbeitsspeicher entfernt. Es steht nun der gesamte Speicher zum Editieren des Programms zur Verfügung. Ferner besteht ein Unterschied im Befehlsumfang. Im Resident-Modus kann nicht auf beliebige Dateien zugegriffen werden. Außerdem sind einige Befehle nicht vorhanden.

Oxford-Pascal läuft auf dem C 64 zusammen mit einem Diskettenlaufwerk 1541 oder einer 4040-Floppystation mit einer speziellen Schnittstelle (Interpod von OCSS). Oxford-Pascal verfügt über einige Erweiterungen gegenüber dem Standard-Pascal. Dies sind im wesentlichen hexadezimale Zahlen, Zugriff auf Arbeitsspeicher und Bildschirm, Verkettung von Programmen, eine Schnittstelle zur Maschinensprache, Grafikbefehle und ein Interface zur Uhr des C 64.

Profi-Pascal besitzt nur einen Modus, dafür aber ein eigenes Betriebssystem — Pascal-DOS — mit mehreren Vorteilen. Dazu gehören der viermal schnellere Diskettenzugriff und bequeme Menüsteuerung. Auch Profi-Pascal wurde um viele Extras erweitert. Assembler-Programme können in Pascal eingebettet werden. Verketten und Segmentieren (Aufteilen eines Programms in mehrere nachladbare Teile) ist möglich. Vorteilhaft ist auch der Zugriff auf den gesamten Arbeitsspeicher des C 64.

Profi-Pascal läuft mit einem Diskettenlaufwerk 1541, besser sind allerdings zwei Laufwerke.

### Streng im Aufbau

Ein Pascal-Programm besteht aus einer Kopfzeile und einem Block. Der Block enthält den Definitions- und Anweisungsteil. Listing 1 zeigt ein Pascal-Programm zur Berechnung des Kreisumfangs. Zum Vergleich wurde das gleiche Problem in Basic formuliert (Listing 2). Die erste Programmzeile ist die Kopfzeile. Sie besteht aus dem reservierten Wort PROGRAM und dem Programmnamen. Die darauf folgenden Zeilen bilden den Programmablauf. Der Definitionsteil be-

steht im Beispiel lediglich aus einer Zeile zur Definition von Variablen. Es ist eine Besonderheit von Pascal, daß sämtliche Variablen, Konstanten, Labels, Dateien und Typen definiert werden müssen. Im Beispielprogramm werden die Variablen R und U definiert. Bei einer Variablendefinition wird einem Variablenamen ein einfacher Datentyp zugewiesen. So einfach ist dies deshalb, weil sich aus einem solchen Typ durch Definition weitere, komplizierte Typen wie Felder gewinnen lassen.

Pascal kennt vier einfache Datentypen: Integer, Real, Boolean und Char. Der Wertbereich der Integer-Zahlen liegt zwischen -32767 (Profi-Pascal) beziehungsweise -32768 (Oxford-Pascal) und 32767. Real-Zahlen werden im Bereich  $\pm 3.4028236 \cdot 10^{38}$  dargestellt. Auch hierbei besteht wieder ein kleiner Unterschied zwischen Profi-Pascal und Oxford-Pascal: Oxford-Pascal verwendet lediglich neun Stellen Genauigkeit, während Profi-Pascal 11 Stellen nutzt.

Variablen des Datentyps Boolean können nur die Werte TRUE und FALSE annehmen. Sie werden zur Darstellung von logischen Werten benutzt. Elemente des Datentyps »CHAR« sind der gesamte Zeichensatz des Computers. Zu beachten ist, daß eine Variable des Typs Char jeweils nur ein Zeichen enthält. Hier noch einige weitere Beispiele zur Variablenvereinbarung:

VAR RADIUS, UMFANG: REAL;

I,J: INTEGER;

B: BOOLEAN;

X,Y:REAL;

ZEICHEN: CHAR;

Eine Variablenvereinbarung beginnt also grundsätzlich mit dem reservierten Wort »VAR«, gefolgt von Variablendefinitionen, die durch Strichpunkte voneinander getrennt werden. Variablen vom gleichen Typ lassen sich in einer Definition, getrennt durch Kommas, zusammenfassen.

Reservierte Wörter spielen in Pascal eine wichtige Rolle. Sie dürfen nicht für Namen verwendet werden. Bild 1 enthält eine Liste der reservierten Wörter.

## Standardprozeduren zur Ein-/Ausgabe

Das Programm Kreis fordert den Benutzer auf, den Radius einzugeben. Zur Ausgabe von Daten auf den Bildschirm sind die beiden Standardprozeduren WRITE und WRITELN vorgesehen. WRITELN ist dabei die Abkürzung von Write Line. WRITELN gibt Daten aus und führt anschließend einen Zeilenvorschub aus. Bei beiden Prozeduren sind die Daten in Klammern anzugeben. Text wird dabei in Hochkommas („, aber nicht“)

gesetzt. Kommas trennen die auszugebenden Daten. Variablen können auch formatiert ausgegeben werden. Im zweiten Aufruf von WRITELN wird die Ausgabe so formatiert, daß der gesamte Ausdruck einschließlich Dezimalpunkt sechs Stellen lang ist. Die Anzahl der Nachkommastellen beträgt zwei.

READ und READLN sind die Standardprozeduren zur Eingabe von Daten über die Tastatur. Die in den Klammern stehenden Variablen müssen vom Typ »INTEGER«, »REAL«, »CHAR« oder »STRING« (wird noch beschrieben) sein. Für jede Variable muß ein Eingabewert vorhanden sein. Zugleich muß der Datentyp des Eingabewerts mit dem Typ der Variablen übereinstimmen. Die Eingabewerte müssen entweder durch ein Leerzeichen oder einen Zeilenwechsel getrennt werden. Andernfalls meldet das System einen Laufzeitfehler.

Eine Ausnahme bildet der Typ »CHAR«. Eine Variable dieses Typs ist genau ein Zeichen lang. Ein Leerzeichen wird ebenfalls als Eingabewert akzeptiert.

READLN ist die Abkürzung von Read line. READLN liest alle Daten bis zum nächsten Return. Den Eingabedaten muß also bei READLN ein Return folgen, falls noch weitere Aufrufe von READ oder READLN innerhalb des Programms auftreten.

Zur Berechnung des Kreisumfangs benötigen wir eine Variable von Typ »REAL«. Ehe die Variable U ausgegeben wird, bekommt

sie einen Wert zugewiesen. Der Umfang wird mit der folgenden Programmzeile berechnet:

$U := 2 * 3.14 * R;$

Das Symbol »:=« wird in Pascal als Wertzuweisung bezeichnet. Auf der linken Seite dieses Operators steht eine Variable und rechts davon ein Ausdruck. Variablen und Ausdrücke müssen immer vom gleichen Typ sein. Eine Ausnahme bilden lediglich Integer-Ausdrücke. Sie dürfen einer Real-Variable zugewiesen werden, da hier kein Informationsverlust auftreten kann. Variablen innerhalb eines Ausdrucks müssen bereits einen Wert erhalten haben. Im Gegensatz zu Basic werden sie am Anfang nicht mit 0 initialisiert, sondern besitzen den zufälligen Wert des Speicherinhalts.

Pascal-Programme werden völlig formatfrei eingegeben. Beide Editoren arbeiten zeilenorientiert und benutzen deshalb Zeilennummern, die jedoch innerhalb des Programms keine Bedeutung haben. Pascal-Anweisungen dürfen sich über mehrere Zeilen erstrecken. Umgekehrt dürfen sich in einer Zeile mehrere Anweisungen befinden. Einzelne Anweisungen werden durch Strichpunkte getrennt. Am Ende eines Pascal-Programms muß ein Punkt stehen.

Der nächste Teil des Pascal-Kurses beschäftigt sich mit dem Editieren von Programmen und den Anweisungen von Pascal.

(Sylvia Gutschmid/Anton Gruber/cg)

```
PROGRAM KREIS;
VAR R, U: REAL;
BEGIN
  WRITELN ('Bitte Radius eingeben');
  READLN(R);
  U := 2 * 3.14 * R;
  WRITELN ('Umfang = ', U, 6:2)
END.
```

**Listing 1. Pascal-Programm zum Berechnen des Kreisumfangs**

```
10  REM KREISUMFANG
20  INPUT »Bitte Radius eingeben«; R
30  LET = 2 * 3.14 * R
40  PRINT »Umfang = «, U
50  END
```

**Listing 2. Berechnung des Kreisumfangs in Basic**

AND	DO	FUNCTION	NIL	PROGRAM	TYPE
ARRAY	DOWNTO	GOTO	NOT	RECORD	UNTIL
BEGIN	ELSE	IF	OF	REPEAT	VAR
CASE	END	IN	OR	SET	WHILE
DIV	FOR	MOD	PROCEDURE	TO	

PACKED ist in Profi-Pascal nicht verfügbar

**Bild 1. Reservierte Wörter in Oxford-Pascal**

Profi-Pascal für C 64, Data Becker, 198 Mark

Profi-Pascal besitzt den vollen Pascal-Wort-Standard. Der Befehlssatz wurde stark erweitert. Schneller Diskettenzugriff (1250 Byte pro Sekunde); der erzeugte Code läuft relativ schnell. Negativ ist, daß die Quelldatei beim Compilieren auf der Systemdiskette gespeichert sein muß (schnelle Abnutzung der Diskette).

Oxford-Pascal, Vertrieb in Deutschland durch Gepo Soft, Gertrudenstr. 37, 4220 Dinslaken, 191,50 Mark

Der Compiler erkennt den vollständigen Standard-Sprachumfang. Programme und Komplikate stehen im Speicher. Das Laden und Compilieren ist recht langsam, aber zum Programmieren steht viel freier Speicherplatz (13 beziehungsweise 32 KByte) zur Verfügung.

Pascal mit dem C 64, Buch mit Software, 250 Seiten Kurs und Programme, Compiler und ausführliche Dokumentation, Markt & Technik Buchverlag, Bestellnr. MT90222, 52 Mark

Der Compiler akzeptiert den gesamten Standard-Sprachumfang mit einigen Erweiterungen und arbeitet recht schnell. Der Full-Screen-Editor ist ungewöhnlich komfortabel. Im Buch findet sich neben einer ausführlichen Bedienungsanleitung ein vollständiger Pascal-Einführungskurs.

Turbo Pascal für den C 128 unter CP/M, Heimsoeth Software, Fraunhofer Str. 13, 8000 München 5, 198 Mark und Markt & Technik Software Verlag, Tel. (089) 4613-220

Turbo-Pascal besitzt einen erweiterten Wortstandard. Quellprogramm und Komplikat werden im Speicher abgelegt. Die Bedienung ist komfortabel und der Compiler selbst ist sehr schnell. Der erzeugte Code ist mit Abstand schneller als unter Profi-Pascal.