

- **Wiederholte Programmausführung**
- Die Wiederholungsanweisungen von Pascal
- Geschachtelte Programmschleifen
- Syntaxdiagramme

Lesen Sie den Begleittext Kapitel 4

## Die Notwendigkeit für wiederholte Programmausführungen

- Aufgabe: Die Quadratwurzel einer positiven Zahl berechnen
- Einschränkung: Es stehen nur arithmetische Operationen zur Verfügung:  $*$ ,  $/$ ,  $+$ ,  $-$

- Lösung: Newton-Verfahren zur Bestimmung von Nullstellen

$$x_0 = 1 \quad \text{und} \quad x_{n+1} = \frac{x_n}{2} + \frac{c}{2x_n}$$

$$\lim_{n \rightarrow \infty} x_n = \sqrt{c}$$

### Beispiel: $c = 2$

$$x_0 = 1$$

$$x_1 = 1/2 + 2/2 = 1.5$$

$$x_2 = 1.5/2 + 2/3 = 1.4166$$

$$x_3 = 1.4166/2 + 2/2.8333 = 1.4141$$

und so weiter . . .

Aufgaben für die Implementierung dieses Verfahrens:

1. Darstellung der  $x_n$
2. Wiederholung der Bildungsvorschrift
3. Abbrechen der Wiederholung

## Variablen, Anweisungen, Schleifen

Die mathematischen Variablen  $x_n$  stellen wir durch die Variable

```
var x: real;
```

dar und realisieren die Bildungsvorschrift für  $x_{n+1}$  mit

```
x := x/2.0 + c/(2.0*x);
```

Diese Anweisung muss für  $n = 1, 2, 3, \dots$  ausgeführt werden. Wir müssen sie also für jedes  $n$  wiederholen.

Pascal bietet verschiedene Möglichkeiten dies zu tun:

**for**-Anweisung, **while**-Anweisung, **repeat**-Anweisung

- Wiederholte Programmausführung

## • Die Wiederholungsanweisungen von Pascal

- Bedingungen zum Abbruch von Programmschleifen
- Geschachtelte Programmschleifen
- Syntaxdiagramme

## Die verschiedenen Wiederholungsanweisungen von Pascal

```

for Anfangswert to Endwert do Anweisungsfolge
for Anfangswert downto Endwert do Anweisungsfolge
while Bedingung do Anweisungsfolge
repeat Anweisungsfolge until Bedingung
  
```

Die for-Schleife führt eine vorgegebene Anzahl Wiederholungen aus.

Die Anzahl Wiederholungen der while- und repeat-Schleife wird durch Aktionen innerhalb der Schleife gesteuert.

## Struktur der for-Anweisung

Kontrollvariable (muss deklariert sein)

```

for Variable:= Ausdruck1 to Ausdruck2 do Anweisung
  
```

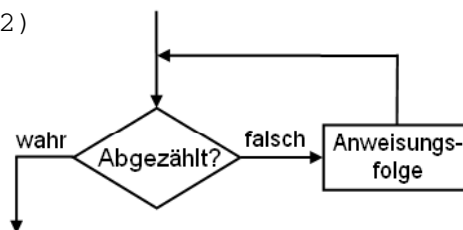
Die Anweisung wird wiederholt solange ausgeführt wie der Wert der Kontrollvariablen kleiner als oder gleich dem Wert von *Ausdruck2* ist.

Beispiel:

```

x:= 5;
for i:= 0 to x do write(i*2)
  
```

Output: 0 2 4 6 8 10



## Struktur der while-Anweisung

```

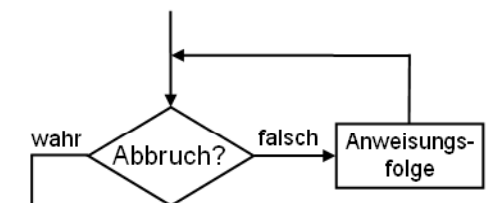
while boolescher Ausdruck do Anweisung
  
```

Die Anweisung wird wiederholt solange ausgeführt wie eine Auswertung des booleschen Ausdrucks den Wahrheitswert "true" ergibt.

```

Beispiel:   i:= 0;
while i < 10 do
begin
    write(i);
    i:= i + 1;
end;
  
```

Output: 0 1 2 3 4 5 6 7 8 9



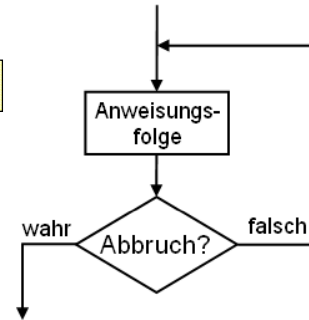
## Struktur der repeat-Anweisung

**repeat** *Anweisung* **until** *boolescher Ausdruck*

Führe Anweisung einmal aus und wiederhole solange bis eine Auswertung des booleschen Ausdrucks den Wahrheitswert "true" ergibt.

Beispiel: `i := 0;`  
**repeat** `write(i); i := i + 1`  
**until** `i = 10`

Output: 0 1 2 3 4 5 6 7 8 9



## Newton-Verfahren mit einer while-Schleife

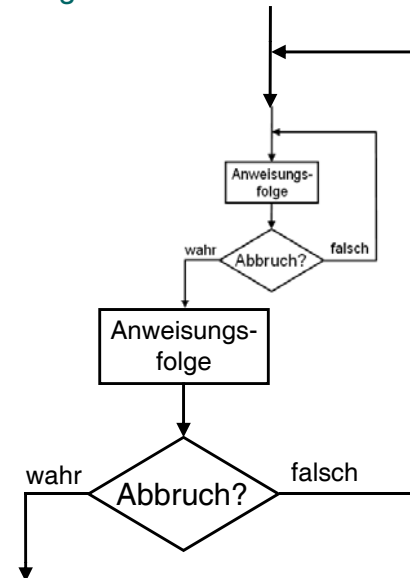
```
program qwurzel;
var c, x: real;
begin
    x := 1.0;
    write('Wurzel aus: ');
    readln(c);
    while abs(x*x - c) > 1.0E-6 do
        x := x/2.0 + c/(2.0*x);
        writeln('Ergebnis = ', x:8:6);
        readln
end.
```

Wiederholung der Bildungsvorschrift

Die Schleife wird abgebrochen, wenn  $|x_n^2 - c| \leq \frac{1}{1000000}$

- Wiederholte Programmausführung
- Die Wiederholungsanweisungen von Pascal
- **Geschachtelte Programmschleifen**
- Syntaxdiagramme

## Zwei geschachtelte Schleifen



• • • • •

• •

Anzahl Durchläufe multiplikativ

## Schachtelung am Beispiel der for-Anweisung

```
for i:= 1 to 3 do
  for j:= 1 to 10 do write(i*j)
```

```
for i:= 1 to 3 do
  begin
    for j:= 1 to 10 do write(i*j);
    writeln;
  end;
```

## Schachtelung am Beispiel der while-Anweisung

Quersumme solange berechnen bis ein "q" eingegeben wird

```
var zahl: string;
```

```
read(zahl);
```

```
while zahl <> 'q' do
```

```
begin
```

```
  summe:= 0;
```

```
  i:= length(zahl);
```

```
  for j:= 1 to i do
```

```
    summe:= summe + ord(zahl[j]) - 48;
```

```
  writeln('Quersumme von ', zahl, ': ', summe)
```

```
end;
```

- Wiederholte Programmausführung
- Die Wiederholungsanweisungen von Pascal
- Geschachtelte Programmschleifen

## • Syntaxdiagramme

## Grundlagen der Syntaxdiagramme

### Elemente der Syntaxdiagramme

Nicht-Terminalsymbole:

Bezeichner

Terminalsymbole:

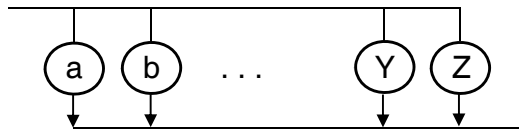
;

Nicht-Terminalsymbole können durch andere Nicht-Terminalsymbole oder durch Terminalsymbole nach vorgegebenen Syntaxregeln ersetzt werden.

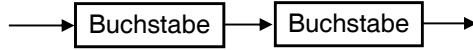
## Grundlagen der Syntaxdiagramme

Produktionsregeln (gelten für Terminal- und Nicht-Terminalsymbole)

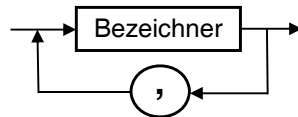
Auswahl:



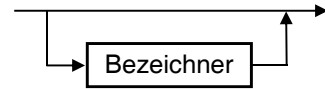
Folge:



Wiederholung:

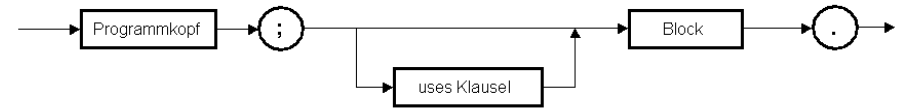


Option:

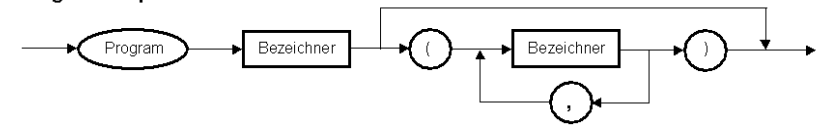


## Syntaxregeln am Beispiel "Programm"

Programm

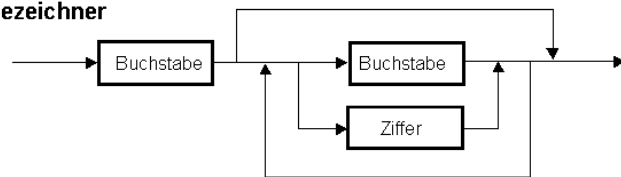


Programmkopf



## Syntaxregeln am Beispiel "Bezeichner"

Bezeichner



Ziffer

