

Grundstruktur Alternative

1. Einseitige Alternative

Syntax: **if** <Bedingung> **then** <Anweisung>;
 wenn dann

2. Zweiseitige Alternative

Syntax:
if <Bedingung> **then** <Dann-Anweisung> **else** <Sonst-Anweisung>;
Wenn **dann** **sonst**

*Merke 1: Vor **else** darf nie ein Semikolon stehen!*

*Merke 2: Bestehen die Dann- oder die Sonst-Anweisung aus mehreren Befehlen, werden diese in **begin** und **end** gesetzt!*

2a. Verschachtelte zweiseitige Alternative

Syntax:
if<Bedingung>**then**<Anweisung>**else**
 if <Bedingung>**then**<Anweisung> **else**<Anweisung>;

3. Die mehrseitige Auswahl

...wird eingesetzt, wenn mehr als 2 Alternativen möglich sind.

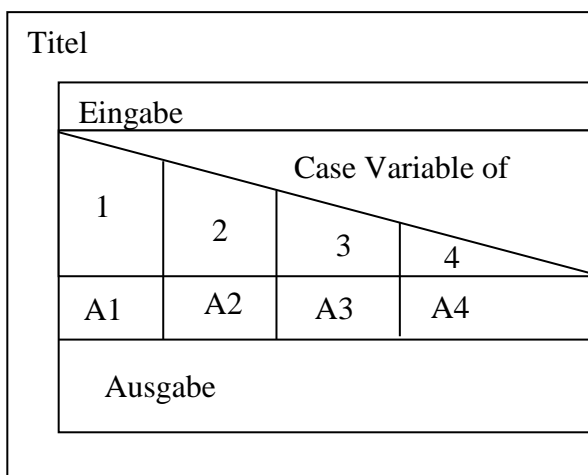
Die case-Anweisung

```
case <Selektor/Variable>of
    <Bereich 1>: Anweisung 1;
    <Bereich 2>: Anweisung 2;
    ...
    <Bereich n>: Anweisung n
else <Sonstanweisung>; end;
```

case wird mit end abgeschlossen

Bereich kann eine Integer-Zahl oder tatsächlich ein Bereich sein.

Struktogramm



Beispiele:
8 einzelner Wert
1..5 von 1 bis 5
7,8 7 und/oder 8

Programmbeispiel:

Rabattgewährung bei TR-Kauf

ab 100 Stück 25 %

ab 50 Stück 20 %

ab 20 Stück 15 %

ab 10 Stück 10 %

Das Programm soll nach Eingabe des Einzelpreises und der Stückzahl den Gesamtpreis, den Rabatt und den tatsächlichen Endpreis ausgeben.

Eingabe?

Welche Variablen werden benötigt?

Benutze folgende Variable:

anz = Stückzahl

p = Einzelpreis

gp = Gesamtpreis

rs = Rabattsatz

r = Rabatt

ep = Endpreis

Rabattberechnung

Eingabe: anz, p				
Gesamtpreis:= anz* p				
Für anz				
1..9	10..19	20..49	50..99	else
rs:=0	.rs:= 0,1	rs:= 0,15	rs:= 0,2	rs:= 0,25
Rabatt:= Gp *rs				
Endpreis:= Gp - r				
Ausgabe: gp; rs; r; ep				

Listing

```
implementation
{$R *.lfm}

{ TForm1 }

procedure TForm1.ButtonkasseClick(Sender: TObject);
var anz:integer;p,gp,rs,r,ep:single;
begin
anz:=StrToInt(Editanz.Text);
p:=StrToFloat(Editp.Text);
gp:= anz*p;

case anz of
1..9:rs:=0;
10..19:rs:=0.1;
20..49:rs:=0.15;
50..99:rs:=0.2;
else rs:=0.25;end;
r:=gp*rs;
ep:=gp-r;
Editrs.Text:=FloatToStrF(rs, ffFixed, 4, 2) ;
Editgp.Text:=FloatToStrF(gp, ffFixed, 10, 2)+'€';
Editr.Text:=FloatToStrF(r, ffFixed, 10, 2)+'€';
Editep.Text:=FloatToStrF(ep, ffFixed, 10, 2)+'€';
end;

end.
```

// Dezimalpunkt setzen
0,1 bedeutet 10% //

// FloatToStrF bedeutet Formatierung
ffFixed liefert feste Stellenzahl
10,2 bedeutet insgesamt 10 Stellen mit
2 Dezimalstellen //

// +'Text' bedeutet, dass an die Ausgabe noch ein
Text, der in Hochkommas steht, angehängt wird //