

VERZWEIGUNGEN UND SPRÜNGE

IF . . . THEN . . . ELSE . . . ;

Beispiele: **if** a>=30 **then** Note:='Ausgezeichnet!';

```
if a>23 then Ergebnis:='Bestanden!'
      else Ergebnis:='nicht bestanden!';
```

```
if a>b then
      begin
        c:=a;
        a:=b;
        b:=c;
      end;
```

```
if a>b then
      begin
        c:=a;
        a:=b;
        b:=c;
      end
      else
      begin
        a:=a-10;
        b:=b+10;
      end;
```

Beachten Sie: Hinter dem **if** muß eine Aussage stehen, die man mit wahr oder falsch beantworten kann. meistens wird das eine Ungleichung oder eine Gleichung sein.

Mehrere Anweisungen nach dem **then** bzw. **else** müssen mit **begin** und **end** zusammengefaßt werden.

Beachten Sie die Semikolon!

Verzweigungen, wie die **if**-Anweisung oder die **Case**-Anweisung der nächsten Seite, sowie alle Schleifentypen können beliebig ineinander verschachtelt werden!

Achten Sie durch sorgfältige Darstellung (Einrücken!) darauf, dass die Übersicht gewahrt bleibt!

CASE . . . OF . . . (ELSE) . . . END;

Mit dieser Verzweigungsanweisung lassen sich oft eine Reihe aufeinanderfolgender oder ineinander verschachtelter **if** – Anweisungen vermeiden. Vorbedingung ist allerdings, daß der Ausdruck in der Case-Anweisung von ordinalem Typ ist; d. h. man muss ihn für eine Aufzählung brauchen können, bzw. er muss vom Typ Integer oder Char sein.

Beispiel: c sei eine Variable des Typs Char.

```
Case c of
  'A': s:='Sie haben ein grosses A eingegeben!';
  'a': s:='Sie haben ein kleines a eingegeben!';
  'B': s:='Sie haben ein grosses B eingegeben!';
  'b': s:='Sie haben ein kleines b eingegeben!';
else
      s:='Sie haben ein anderes Zeichen eingegeben!';
end;
```

Mit **if** könnte dasselbe Programm wie folgt aussehen:

```
if c='A' then s:='Sie haben ein grosses A eingegeben!';
if c='a' then s:='Sie haben ein kleines a eingegeben!';
if c='B' then s:='Sie haben ein grosses B eingegeben!';
if c='b' then s:='Sie haben ein kleines b eingegeben!';
      else s:='Sie haben ein anderes Zeichen eingegeben!';
```

Besonders vorteilhaft ist bei der Anwendung von Case, dass man sehr einfach Bereiche benützen kann:

Beispiel: Punktzahl sei eine Variable des Typs Integer.

```
Case Punktzahl of
  0.. 9:      Zensur:='schwache Leistung';
  10..19:    Zensur:='ungenügend';
  20..29:    Zensur:='genügend';
  30..39:    Zensur:='gut';
  40..45:    Zensur:='sehr gut';
  -10..-1, 46..100: Zensur:='diese Punktzahlen existieren gar nicht!';
end;
```

GOTO, CONTINUE, BREAK UND EXIT

Diese Bedingungen sollten in Delphi möglichst vermieden werden.