

Programmieren 1

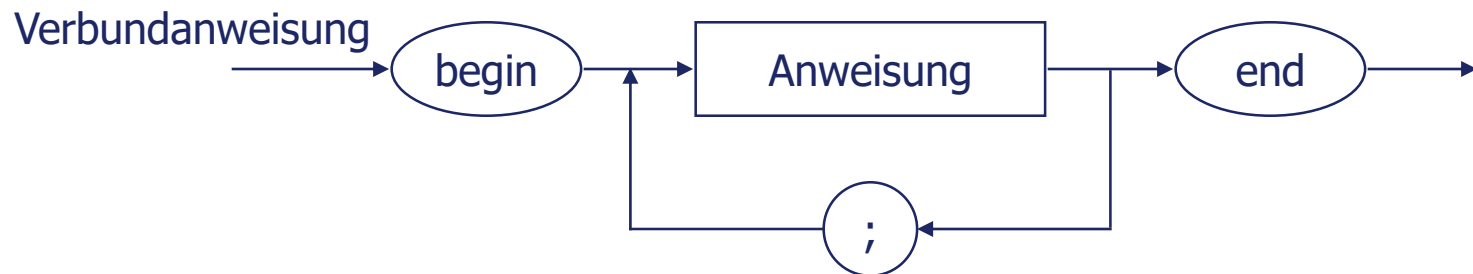


Wintersemester 2016/2017

Marcus Riemer, B.Sc.

Basierend auf den Unterlagen „Programmstrukturen 1“
von Prof. Dr. Andreas Häuslein

- Eine Verbundanweisung (zusammengesetzte Anweisung) fasst eine beliebige Anzahl von Anweisungen zu einer Anweisung zusammen
- Eingebettete Anweisungen werden in der Reihenfolge, in der sie in der Verbundanweisung enthalten sind, ausgeführt (Sequenz)
- Syntaktischer Aufbau:



- Eine Verbundanweisung gilt syntaktisch als *eine* Anweisung -> sie darf überall dort stehen, wo syntaktisch nur eine Anweisung zugelassen ist

- Bekannte Verwendung der Verbundanweisung: Programmrumpf

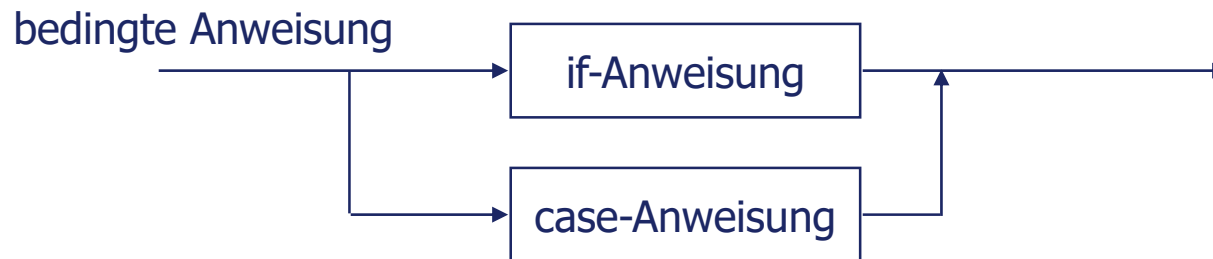
```
program Zweites;  
  
var  
    Menge : Cardinal;  
    Preis : Real;  
    Gesamtbetrag : Real;  
begin  
    readln (Menge);  
    readln (Preis);  
    Gesamtbetrag := Menge * Preis;  
    writeln (Gesamtbetrag)  
end
```

**Verbund-
anweisung**

- Weitere Verwendung von Verbundanweisungen:
 - Als Bestandteil von anderen strukturierten Anweisungen
 - In Prozedur-/Funktionsdeklarationen

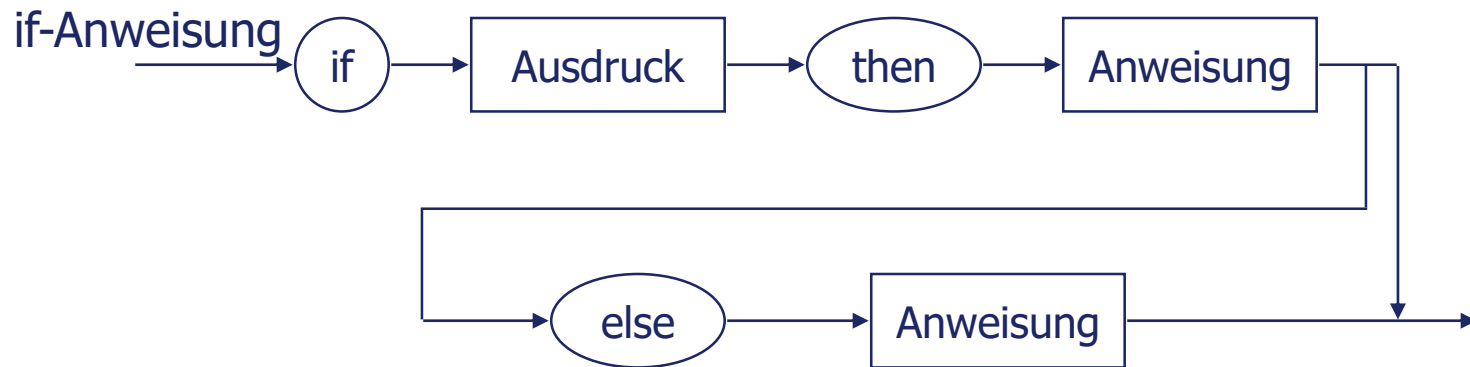
- Bedingte Anweisungen dienen dazu, die algorithmische Struktur *Selektion* in Programmen umzusetzen
- Bedingte Anweisungen beinhalten Anweisungen, von denen keine oder eine bestimmte ausgeführt wird
- Die Auswahl der evtl. auszuführenden Anweisung wird durch Bedingungen gesteuert

- Syntaktische Unterscheidung:



- Die if-Anweisung realisiert die algorithmischen Strukturen *bedingte Anweisung* und *Alternative*
- Die case-Anweisung realisiert die algorithmische Struktur *Mehrfach-Auswahl*

Syntaktischer Aufbau:



Hinweise:

- Der Ausdruck stellt die Bedingung dar, er muss ein *boolescher* Ausdruck sein
- Für die nicht-terminalen Symbole <Anweisung> können Anweisungen beliebiger Art eingesetzt werden (auch wieder if-Anweisungen)
- Vor dem Schlüsselwort `else` darf kein Semikolon stehen!



- Verhalten beim Programmlauf
 - Wenn die Auswertung des booleschen Ausdrucks **True** ergibt,
 - Wird die Anweisung im then-Teil ausgeführt
 - Danach wird die if-Anweisung verlassen und der Programmlauf mit der Anweisung nach der if-Anweisung fortgesetzt
 - Insbesondere wird ein evtl. vorhandener else-Teil übersprungen
 - Wenn die Auswertung des booleschen Ausdrucks **False** ergibt,
 - werden, wenn ein else-Teil vorhanden ist, wird die Anweisung im else-Teil ausgeführt
 - Danach wird die if-Anweisung verlassen
 - wird, wenn kein else-Teil vorhanden ist, die if-Anweisung sofort nach Auswertung des booleschen Ausdrucks verlassen



Beispiele:

```
if Temperatur > 30 then writeln ('Hitzefrei!');
```

```
if PunkteZahl < 50 then  
    writeln ('Leider Pech gehabt!')  
else  
    writeln ('Herzlichen Glückwunsch!');
```

```
if Lagerbestand < Bestellmenge then  
    Ausliefermenge := Lagerbestand  
else  
    Ausliefermenge := Bestellmenge;
```

```
if Lagerbestand < Bestellmenge then  
    begin  
        Ausliefermenge := Lagerbestand;  
        Lagerbestand := 0;  
    end  
else  
    begin  
        Ausliefermenge := Bestellmenge;  
        Lagerbestand := Lagerbestand - Bestellmenge;  
    end;
```



- Weitere Beispiele:

```
if N <> 0 then
  begin
    Mittelwert := Summe / N;
    writeln ('Mittelwert: ', Mittelwert);
  end
else
  writeln ('Fehler: Division durch Null');
```

Boolesche Variable als Bedingung:

```
var Schaltjahr : Boolean;
...
Schaltjahr := ...
...
if Schaltjahr then
  FebruarMax := 29
else
  FebruarMax := 28;
...
```


Weitere Beispiele:

Boolesche Ausdrücke mit booleschen Operatoren als Bedingung:

```
if not Schaltjahr then  
    FebruarMax := 28  
else  
    FebruarMax := 29;
```

```
if not Volljaehrig and not Sondergenehmigung then  
    writeln ('Zulassung verweigert');
```

```
if WareBezahlt or KundeBekannt and (Bestellwert < 10) then  
    writeln ('Ware ausliefern');
```

```
if LagerLeer and (Bestellmenge > 0) then  
    Rueckstand := Rueckstand + Bestellmenge;
```

```
if TankLeer and (Flughoehe > 0) and not istSegelflugzeug then  
    writeln ('Umgehend landen!');
```



- Geschachtelte if-Anweisungen

- Dienen zur Auswahl unter mehr als 2 Anweisungen
- Prinzip: Anweisungen im (then- bzw.) im else-Teil einer if-Anweisung können wiederum if-Anweisungen sein:

```
if B1 then S1
else      if B2 then S2
          else  if B3 then S3
                else ...
```

- Eine geschachtelte if-Anweisung gilt als syntaktisch als *eine* Anweisung
- Vorsicht: Bei if-then-else-Anweisung im then-Teil einer if-then-Anweisung ergibt sich Mehrdeutigkeit:

```
if B1 then  if B2 then S2 else S3;
```

Regel: else-Zweig gehört immer zur letzten Bedingung ohne else-Zweig



- Geschachtelte if-Anweisungen
 - Beispiel

```
write ('Größte Zahl: ');  
if Zahl1 > Zahl2 then  
    if Zahl3 > Zahl1 then  
        writeln (Zahl3)  
    else  
        writeln (Zahl1)  
else  
    if Zahl3 > Zahl2 then  
        writeln (Zahl3)  
    else  
        writeln (Zahl2);
```

- **Beispiel: Preisberechnung mit gestaffeltem Rabatt**

```
program Sechstes;

const
  MWST_SATZ = 19;
  RABATT_1   = 3;
  RABATT_2   = 5;
  RABATT_3   = 10;

var
  Menge : Cardinal;
  Preis  : Real;
  Rabatt, Nettobetrag, Bruttobetrag : Real;

begin
  write ('Bitte geben Sie die Menge ein: ');
  readln (Menge);
  write ('Bitte geben Sie den Preis ein: ');
  readln (Preis);
  ...{Fortsetzung nächste Seite}
```

{Fortsetzung Programm}

```
Nettobetrag := Menge * Preis;
```

```
Rabatt := 0;
```

Möglichkeit 1

```
if Nettobetrag >= 1000 then
```

```
    Rabatt := Nettobetrag * RABATT_1/100
```

```
else
```

```
    if Nettobetrag >= 500 then
```

```
        Rabatt := Nettobetrag * RABATT_2/100
```

```
    else
```

```
        if Nettobetrag >= 200 then
```

```
            Rabatt := Nettobetrag * RABATT_3/100
```

X

```
        else
```

```
            Rabatt := 0;
```

Möglichkeit 2

```
Nettobetrag := Nettobetrag - Rabatt;
```

```
writeln ('Der Rabatt beträgt: ', Rabatt);
```

```
writeln ('Der Nettobetrag ist: ', Nettobetrag);
```

```
Bruttobetrag := Nettobetrag + Nettobetrag * MWST_SATZ /100;
```

```
writeln ('Der Bruttobetrag ist: ', Bruttobetrag);
```

```
writeln ('MwSt-Satz: ', MWST_SATZ, '%');
```

```
readln;
```

```
end.
```

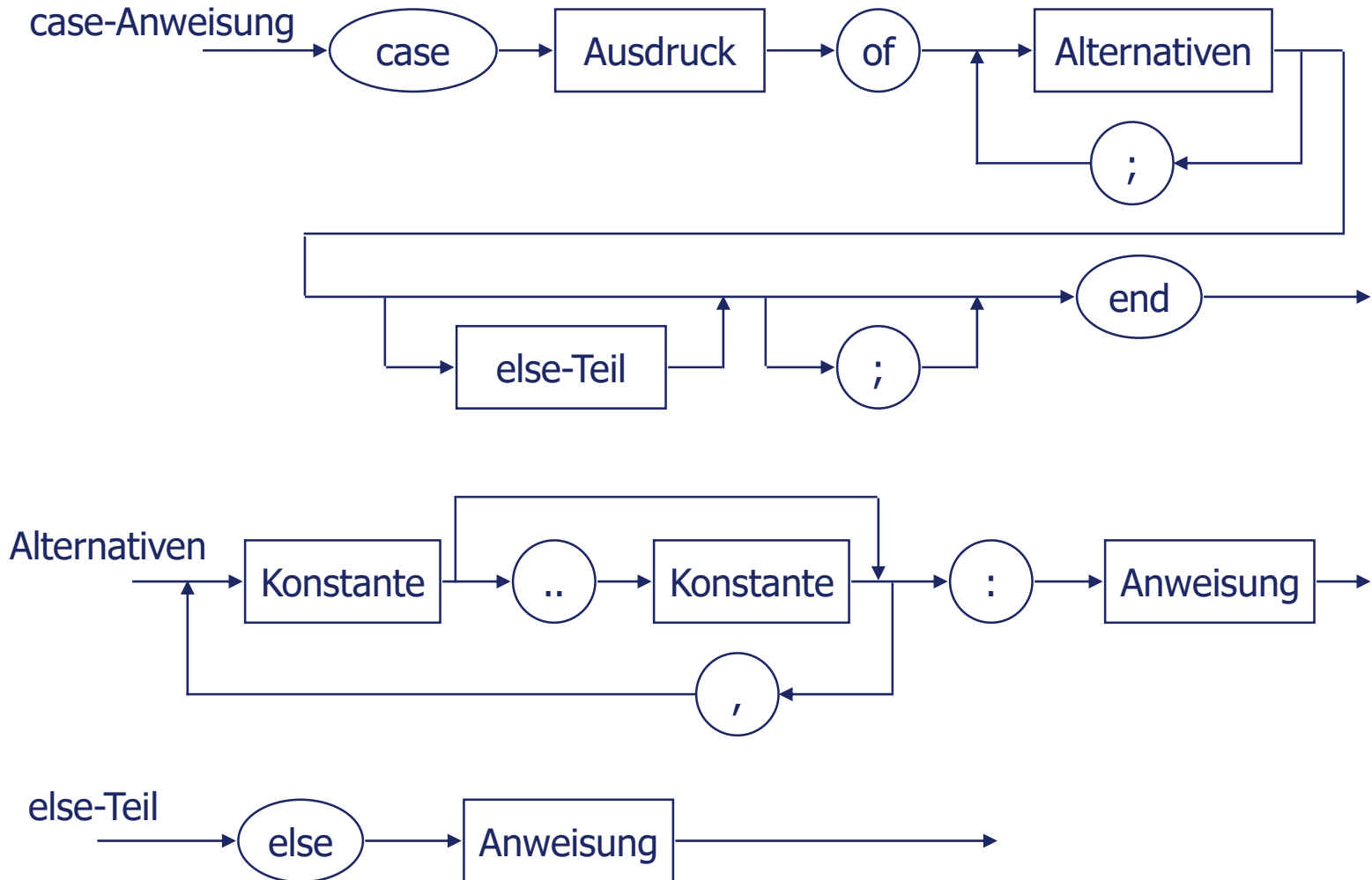
Achtung, Achtung: typischer Fehler!!

Variable `Rabatt` muss in jedem Fall einen Wert erhalten!



- Realisiert die algorithmische Struktur der Mehrfachauswahl (Fallunterscheidung)
- Auswahl unter mehr als 2 Anweisungen
- Übersichtlicher als geschachtelte if-Anweisungen
- Basiert auf Vergleich *eines* Ausdrucks mit unterschiedlichen Konstanten(-bereichen) (impliziter Gleichheitstest)
- Wert des Ausdrucks, der mit Konstanten verglichen wird, muss zu ordinalem Datentyp gehören
- Konstanten und der Wert des Ausdrucks müssen kompatibel sein
- Vergleiche zwischen 2 variablen Werten (Ausdrücke) sind nicht realisierbar

- Syntaktischer Aufbau:



- Beispiel

```
var
  Operator : Char;
  Zahl1, Zahl2, Zahl3 : Integer;

begin
  ...
  readln (Operator);
  case Operator of
    '+' : Zahl3 := Zahl1 + Zahl2;
    '-' : Zahl3 := Zahl1 - Zahl2;
    '*' : Zahl3 := Zahl1 * Zahl2;
    '/' : Zahl3 := Zahl1 div Zahl2
  else
    writeln ('Unzulässiger Operator');
  end;
  ...
end.
```




- Anmerkungen zur Ausführung
 - Der Wert des Selektors wird ermittelt
 - Wert des Selektors wird mit Marken verglichen
 - Die Anweisungen, deren Marke mit dem Wert des Selektors übereinstimmt oder in deren Markenbereich der Wert des Selektors liegt, werden ausgeführt
 - Stimmt keine der Marken mit dem Wert des Selektors überein, wird der else-Teil ausgeführt (sofern vorhanden)
 - Ist kein else-Teil vorhanden und liegt keine Markenübereinstimmung vor, wird die case-Anweisung ohne Ausführung einer Anweisung verlassen



Schreibt einen kleinen Taschenrechner für Ganzzahlen, nimmt dabei die vorherige Folie als Grundlage. Vom Benutzer müssen allerdings vorher noch zwei Werte eingelesen werden. Nehmt dabei die weiteren folgenden Änderungen gegenüber der Vorlage vor:

- Keine Zuweisung an eine Variable `Zahl3`, stattdessen unmittelbare Ausgabe in der `case`-Anweisung.
- Die ganzzahlige Division möchten wir etwas ausführlicher bearbeiten, verwendet an dieser Stelle eine Verbundanweisung und eine `if`-Abfrage mit `else`-Teil:
 - Quittiert den Versuch einer Division durch 0 mit einer aussagekräftigen Fehlermeldung.
 - Gebt das vollständige Ergebnis der Division nach dem folgenden Schema aus:
 - `10 / 5 → 2 Rest 0`
 - `10 / 6 → 1 Rest 4`
 - `10 / 1 → 10 Rest 0`



- Anmerkungen zur Syntax
 - Der Ausdruck zwischen den Schlüsselworten `case` und `of` wird als *Selektor* bezeichnet
 - Die Konstanten, durch die die Fälle unterschieden werden, heißen *Marken* (case-Marken)
 - Soll eine Anweisung für mehrere Werte des Selektors ausgeführt werden, können mehrere Markenwerte bzw. -wertebereiche angegeben werden
 - Alle Marken innerhalb einer case-Anweisung müssen unterschiedlich sein
 - Die case-Anweisung wird durch das Schlüsselwort `end` abgeschlossen



- Weitere Beispiele

```
var Ziffer : Char;  
  
begin  
  ...  
  Ziffer := ...  
  case Ziffer of  
    '0': writeln ('Null');  
    '1': writeln ('Eins');  
    '2': writeln ('Zwei');  
    '3': writeln ('Drei');  
    '4': writeln ('Vier');  
    '5': writeln ('Fünf');  
    '6': writeln ('Sechs');  
    '7': writeln ('Sieben');  
    '8': writeln ('Acht');  
    '9': writeln ('Neun');  
  end;
```

```
var Zahl : byte;  
  
begin  
  ...  
  Zahl := ...  
  case Zahl of  
    0: writeln ('Null');  
    1: writeln ('Eins');  
    2: writeln ('Zwei');  
    3: writeln ('Drei');  
    4: writeln ('Vier');  
    5: writeln ('Fünf');  
    6: writeln ('Sechs');  
    7: writeln ('Sieben');  
    8: writeln ('Acht');  
    9: writeln ('Neun');  
  end;
```



- Weitere Beispiele

```
var Kurs: word;  
  
begin  
  ...  
  Kurs:= ...  
  case Kurs of  
    0..45,  
    316..360: writeln ('Nord');  
    46..135 : writeln ('Ost');  
    136..225: writeln ('Süd');  
    226..315: writeln ('West')  
  else  
    writeln ('Wert ungültig')  
  end;
```

```
var Groesse : Char;  
  
begin  
  ...  
  readln (Groesse);  
  case Groesse of  
    'S': writeln ('Small');  
    'M': writeln ('Medium');  
    'L': writeln ('Large')  
  else  
    writeln ('Unzulässige Angabe')  
  end;
```

- Weiteres Beispiel:

```
var KW_Aktuell, KW_Termin: byte;
```

```
begin
```

```
...
```

```
case KW_Termin - KW_Aktuell of
```

```
  0: writeln ('Termin ist diese Woche');
```

```
  1: writeln ('Termin ist nächste Woche');
```

```
  2..6: writeln ('Termin ist in den nächsten Wochen');
```

```
  7..51: writeln ('Termin ist irgendwann später')
```

```
else
```

```
  writeln ('Termin ist vorbei!');
```

```
end;
```

```
...
```



Selektor darf beliebiger Ausdruck sein

- Weiteres Beispiel:

```
var Zahl1, Zahl2, Zahl3 : integer;  
    Operator : char;
```

```
begin  
    ...  
    Operator := ...
```

```
case Operator of  
    '+': begin  
        Zahl3 := Zahl1 + Zahl2;  
        writeln ('Ergebnis der Addition: ', Zahl3;  
    end;  
    '-': begin  
        Zahl3 := Zahl1 - Zahl2;  
        writeln ('Ergebnis der Subtraktion: ', Zahl3;  
    end;  
    .  
    .  
    .  
end;
```

Mehrere Anweisungen pro Fall
zu Verbundanweisung zusammenfassen



Beispiel zur Demonstration der Übersichtlichkeit

```
if Zeugnisnote = 1 then writeln ('Sehr gut!')
else
  if Zeugnisnote = 2 then writeln ('Gut')
  else
    if Zeugnisnote = 3 then writeln ('Halb gut')
    else
      if Zeugnisnote = 4 then writeln ('Gerade noch')
      else
        if (Zeugnisnote = 5) or (Zeugnisnote = 6) then
          writeln ('Durchgefallen!')
        else writeln ('Note nicht zulässig!');
```

```
case Zeugnisnote of
  1: writeln ('Sehr gut!');
  2: writeln ('Gut');
  3: writeln ('Halb gut');
  4: writeln ('Gerade noch');
  5, 6: writeln ('Durchgefallen!')
else
  writeln ('Note nicht zulässig!');
end;
```


- Beispielprogramm

```
program Siebtes;  
{Ergänzung um einen gestaffelten Rabatt in Abhängigkeit von der  
Menge, realisiert durch eine case-Anweisung}
```

```
const  
  MwSt_Satz = 19;  
  Rabatt1 = 3;  {Rabatt in Prozent}  
  Rabatt2 = 5;  {Rabatt in Prozent}  
  Rabatt3 = 10; {Rabatt in Prozent}
```

```
var  
  Menge : Word;  
  Preis : Real;  
  Rabatt, Nettobetrag, Bruttobetrag : Real;
```

```
begin  
  write ('Bitte geben Sie die Menge ein: ');  
  readln (Menge);  
  write ('Bitte geben Sie den Preis ein: ');  
  readln (Preis);  
  Nettobetrag := Menge * Preis;  
  {Fortsetzung nächste Seite...}
```



```
case Menge of
  1000..high(Menge)    : Rabatt := Nettobetrag * Rabatt3 / 100;
  500..999             : Rabatt := Nettobetrag * Rabatt2 / 100;
  100..499             : Rabatt := Nettobetrag * Rabatt1 / 100
else
  Rabatt := 0
end;
Nettobetrag := Nettobetrag - Rabatt;
writeln ('Der Rabatt beträgt: ', Rabatt);
writeln ('Der Nettobetrag ist: ', Nettobetrag);
Bruttobetrag := Nettobetrag + Nettobetrag *
                MwSt_Satz / 100;
writeln ('Der Bruttobetrag ist: ', Bruttobetrag);
writeln ('MwSt-Satz: ', MwSt_Satz, '%');
end.
```

Ihr möchtet mit mehreren Freunden einen Gutschein für einen Freifilm einlösen, könnt Euch allerdings nicht auf einen der verfügbaren Titel einigen. Schreibt ein Programm, um dieses Problem zu beheben.

- Dazu fragt das Programm zunächst mehrere Eckdaten ab, diese werden in Variablen gespeichert. Weist den Benutzer dabei auf die zulässigen Werte hin (0-255 Tiere, 6-120 Jahre alt).
 - Wie viele Tiere sollen in dem Film mindestens vorkommen?
 - Wie alt ist die jüngste Person in der teilnehmenden Gruppe?
- Nach der Abfrage all dieser Werte erfolgt eine Plausibilitätsprüfung, schließlich ergibt nicht jede Eingabe Sinn. Im Falle einer ungültigen Eingabe wird die auf der nächsten Seite beschriebene Filmauswahl nicht durchgeführt und stattdessen eine Fehlermeldung angezeigt. Nutzt `if`-Verzweigungen mit einem `else`-Teil für die tatsächliche Verarbeitung.
 - Die Anzahl der Tiere ist negativ oder größer als 255
 - Die jüngste Person ist noch nicht sechs oder älter als 120.

- Die folgenden Filme stehen zur Auswahl. Verschachteln Sie zwei case-Anweisungen, wobei das Alter als erstes Kriterium und die Anzahl der Tiere als zweites Kriterium verwendet werden soll. Bedenken sie, dass die Anzahl der Tiere eine Mindest-Anzahl ist, mehr Tiere als gewünscht sind also zulässig, weniger jedoch nicht.
 - "Shaun of the Dead", für Personen ab 18 mit 12 Tieren.
 - "Das Brüllen der Lämmer", für Personen ab 18 mit 4 Tieren.
 - "Verhängnis", für Personen ab 18 mit 3 Tieren.
 - "Säge 14", für Personen ab 18 ohne Tiere.
 - "Jenseits von Wedel" mit 4 Tieren, ab 12 Jahren.
 - "Wedel sehen und sterben", ein toller Film ohne Tiere ab 12 Jahren.
 - "101 Dalmatiner", ab 6 Jahren mit 103 Tieren.
 - "Findet Nemo", ab 6 Jahren mit 14 Tieren.
 - "Susi und Strolch", ab 6 Jahren mit 3 Tieren.
 - "Oben", ab 6 Jahren ohne Tiere.