



Übersicht Strukturen der Programmierung

Sequenz

Die Sequenz ist die einfachste Grundstruktur eines Programms und ist eine lineare, also nicht verzweigende und sich nicht wiederholende Programm-/Anweisungsfolge. (Meist nach EVA-Prinzip). Beispiel:

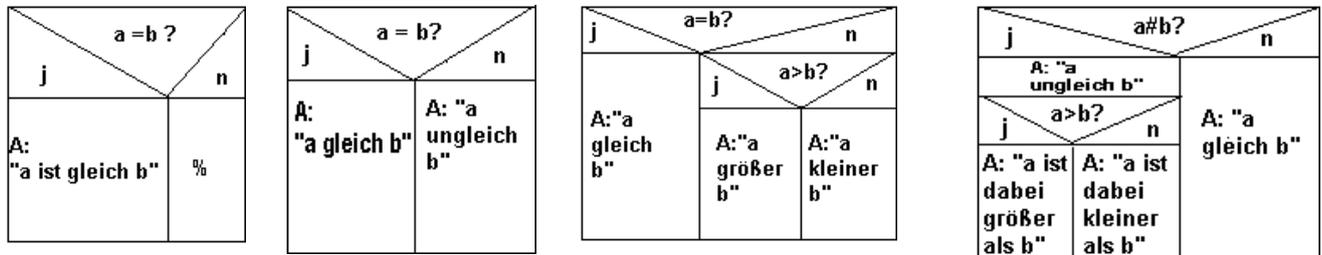
E: a	Eingabe der Größe a durch den User
e=a*2	Verarbeitung (Verdopplung) der Größe a und speichern in der Größe e
A: e	Ausgabe der Größe e

Verzweigungen

In Programmen müssen in Abhängigkeit von Ereignissen Fallunterscheidungen erfolgen. So müssen z.B. zur Fehlervermeidung bei mathematischen Aufgaben bestimmte Eingabefehler abgefangen werden. (z.B.: wenn Divisor = 0, dann Fehlerausgabe)

einfache Verzweigung

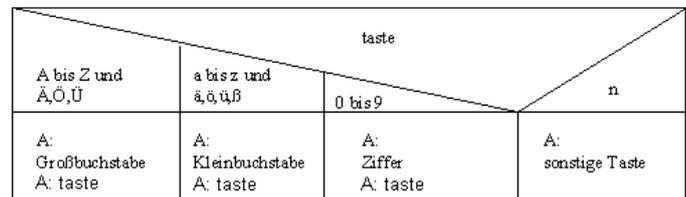
Schon die einfache Verzweigung erlaubt eine sehr differenzierte Reaktion auf verschiedenste Programmsituationen. Syntaxdiagramm der IF .. THEN .. ELSE .. - Anweisung:



Mehrfachverzweigung

In Abhängigkeit vom aktuellen Wert des ordinalen Selektors wird zu der Anweisungsfolge verzweigt, deren Konstante(n)(bereich) dem aktuellen Selektorwert entspricht. Ist der aktuelle Wert nicht in dieser Liste enthalten, wird die Anweisungsfolge von ELSE ausgeführt.

Syntaxdiagramm der CASE .. OF – Anweisung:



Schleifen

Schleifen dienen der Wiederholung von Programmteilen. Damit keine unendlichen Wiederholungen stattfinden, muss jede Schleife unter einer bestimmten Abbruchbedingung enden. Diese Abbruchbedingung muss auch tatsächlich erreicht werden, d.h. die Bedingung muss im Schleifenkörper aufgebaut werden. Die Überprüfung dieser Abbruchbedingung kann vor, nach oder innerhalb der Wiederholung der Programmsequenz erfolgen.

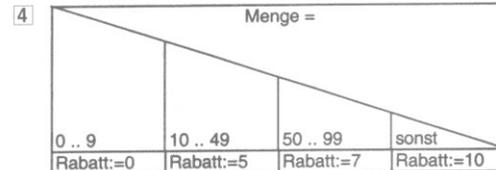
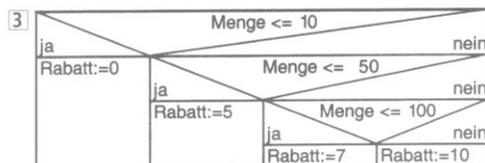
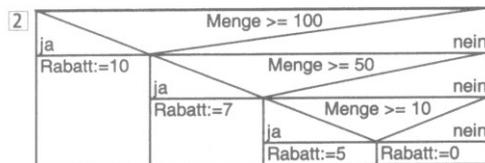
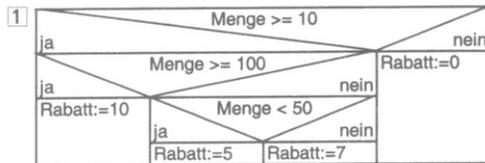
REPEAT...UNTIL (kopfgesteuert) VBA: Do While Bed. Loop	Wiederholt die Anweisungen zwischen REPEAT und UNTIL bis die Abbruchbedingung wahr (TRUE) ist. (Prüfung am Anfang)	
WHILE .. DO (fußgesteuert) VBA: Do ... Loop Until Bed.	Solange die Wiederholungsbedingung wahr (TRUE) ist, werden die Anweisungen zwischen DO und END wiederholt. => Abbruch wenn Bedingung = FALSE (Prüfung am Ende)	
FOR .. TO .. DO (zählergesteuert)	Zählschleife. bei der die Anweisungen zwischen DO und END wiederholt werden. Dabei wird eine Zählvariable mit einem bestimmten Anfangswert um einen bestimmten Betrag erhöht/erniedrigt. Abbruch erfolgt, wenn die Zählvariable den Endwert über-/unterschritten hat.	



Übungsaufgaben Struktogramme

1) Struktogramm „Brauerei“

Eine Brauerei gewährt ihren Kunden bei Mindestabnahme von 10 Kästen 5%, bei Mindestabnahme von 50 Kästen 7% und bei Mindestabnahme von 100 Kästen 10% Rabatt. Diese Regelung soll durch einen Programmbau- stein, z. B. bei der Fakturierung, automatisch berücksichtigt werden. Kontrollieren Sie die gegebenen Strukto- gramme daraufhin, ob sie die gegebene Regelung logisch richtig umsetzen. Prüfen Sie, welches Struktogramm fehlerhaft ist.



5

Tabelle		
Index	Wert1	Wert2
1	100	10
2	50	7
3	10	5
4	1	0

```

Index := 1
solange Menge < Wert1[Index]
(wiederhole) Index := Index + 1
Rabatt := Wert2[Index]
    
```

2) Struktogramm "bmi-rechner"

Entwickeln Sie ein Struktogramm, das den BMI (Body-Maß-Index) des Benutzers berechnet und "einordnet". Der BMI ist ein Maß für das Gewicht in Relation zur Körpergröße, er wird wie folgt berechnet (Gewicht in kg und Körpergröße in cm): $bmi = gewicht * 10000 / groesse^2$

BMI-Index	Frauen	Männer
Untergewicht	<19	<20
Normalgewicht	19 - 24	20 - 25
Übergewicht	25 - 30	26 - 30
Behandlungsbedürftiges Übergewicht	> 30	

3) Struktogramm „Anruf“

Entwickeln Sie ein Struktogramm, das den Ablauf eines Telefongesprächs wiedergibt. Stellen Sie Wiederholungsversuche (falls niemand abnimmt!!) durch unterschiedliche Schleifentypen dar und ermöglichen Sie einen Anrufbeantworter.

4) Struktogramm „Wurzel“

Entwickeln Sie ein Struktogramm, das die Quadratwurzel aus einer positiven ganzen Quadratzahl zieht. Eingabe: ganzzahlige Quadratzahl (z.B. 36). Ausgabe: Wurzel daraus (z.B. 6). Anmerkung: Wenn die eingegebene Zahl keine Quadratzahl ist (z.B. 37), dann soll die Ausgabe eine Fehlermeldung sein („keine Quadratzahl“).

5) Struktogramm „Binärzahlen“

Erstellen Sie ein Struktogramm, das aus einer ganzen, positiven Zahl (0 bis 255) den Binärcode ermittelt. So ergibt die Zahl Null den Binärcode 0000 0000, die Zahl 255 den Binärcode 1111 1111.

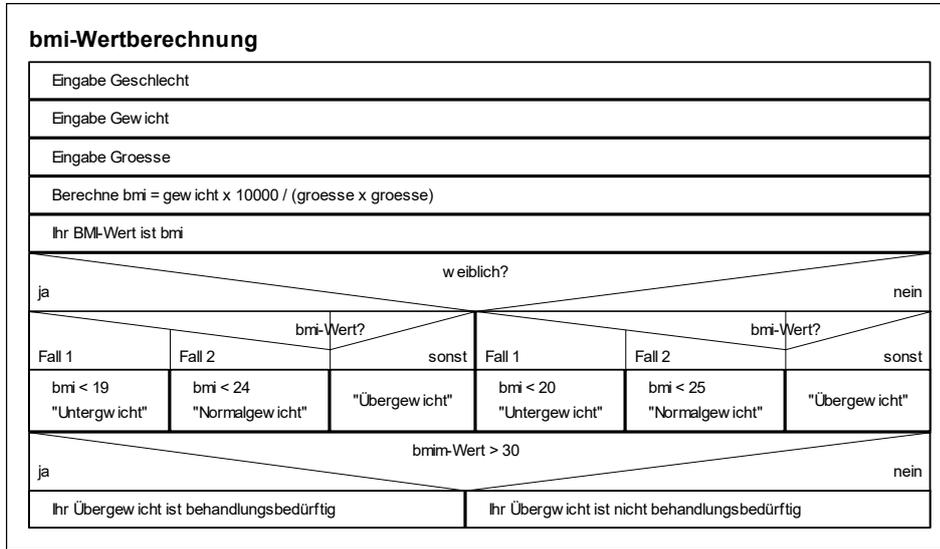
6) Struktogramm „Modula – Restwertmethode“

Erstellen Sie ein Struktogramm, das aus einer ganzen, positiven Zahl eine Division vornimmt, die eine positive ganze Zahl zurückgibt und je nach Modulaverfahren einen Restwert ermittelt. Beispiel **17 Mod 2** ergibt 8 Rest 1, **37 Mod 6** ergibt 6 Rest 1.

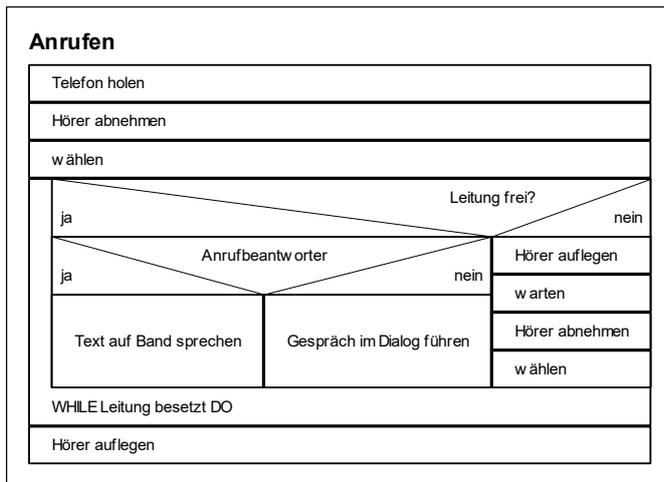


Musterlösung

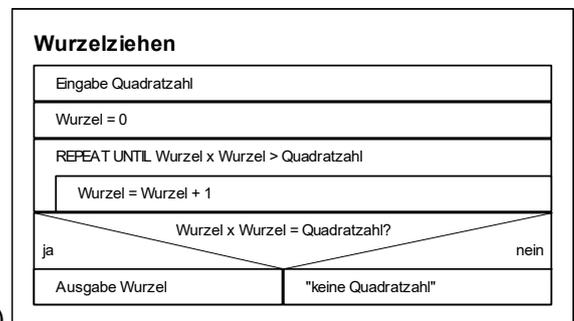
1) Antwort 3 ist falsch (es muss heißen <100, da <= 100 bedeutet, das erst ab 101 Kästen der Rabatt von 10% gewährt wird)



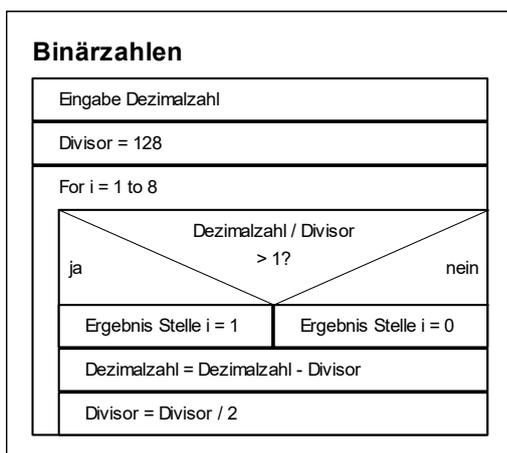
2)



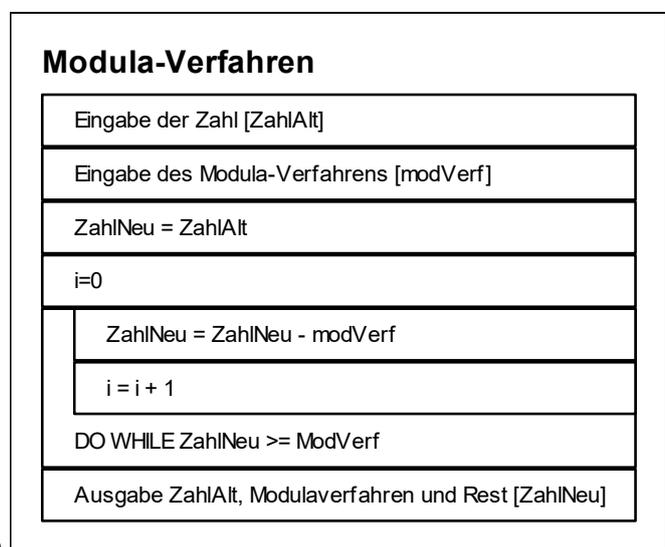
3)



4)



5)



6)

