



## Umsetzung Programmier-Grundstrukturen in MS-Access VBA

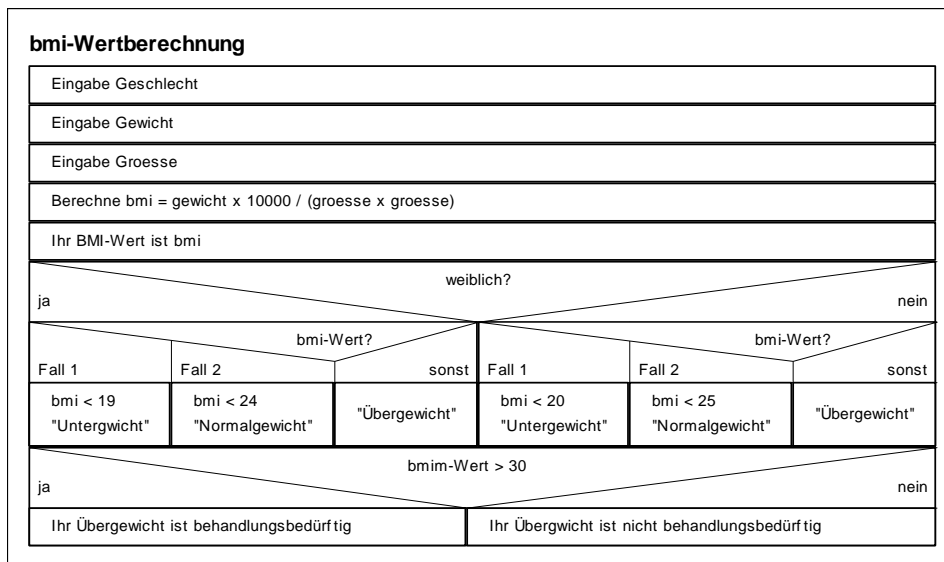
Für die folgenden Übungsbeispiele erstellen Sie je eine Schaltfläche (ohne Assistenten) im Formular „Stundeneingabe“ der Datenbank „PersonaZeitarbeit“. Dem Ereignis „Bei Klick“ fügen Sie den entsprechenden Programmcode zu.

### 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	

Musterlösung des Struktogramms:



#### Hinweis:

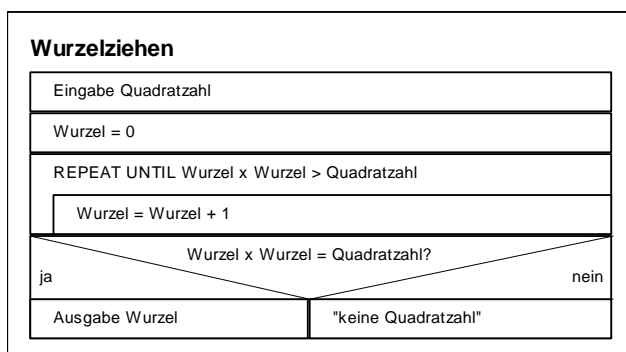
Die benötigten Programmstrukturen (hier: If ... Then, Elself ... End If) entnehmen Sie bitte der Excel-VBA-Reihe oder der Online-Hilfe!!

Die Eingaben erfolgen über drei InputBoxen, die Ausgaben über eine MessageBox.

### 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“).

Musterlösung Struktogramm:



#### Hinweis:

Die benötigten Programmstrukturen sind hier neben der Verzweigung die Do Loop ... Until Schleife.

Mit der Wurzelfunktion kann die genaue Wurzel der Variable „Quadratzahl“ ermittelt werden (Format (Sqr(Quadratzahl), "##,##0.0000"))

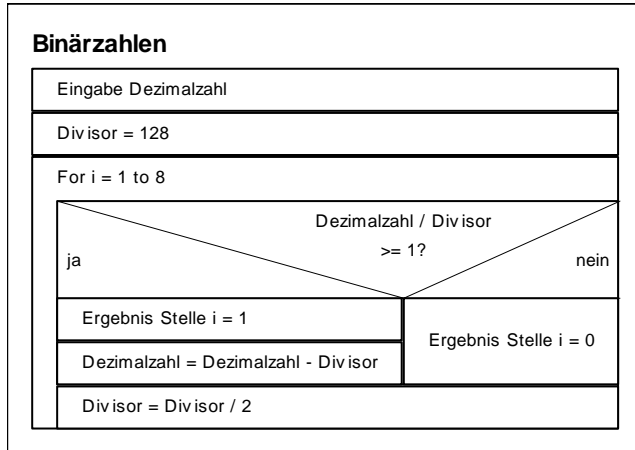
Andere Lösungen sind aber auch denkbar!



## Umsetzung Programmier-Grundstrukturen in MS-Access VBA (2)

### 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.



### Exkurs Datenfeld

Ein Satz aufeinanderfolgender, indizierter Elemente, die den gleichen Datentyp besitzen. Jedes Element eines Feldes wird durch eine eindeutige Index-Nummer identifiziert. Wenn bei einem Feldelement Änderungen vorgenommen werden, bleiben die anderen Elemente davon unbeeinflusst.

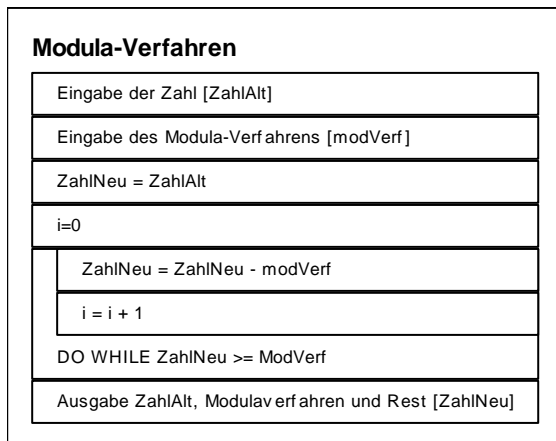
Der Variable Stelle für die unterschiedlichen Binärzahlen kann durch die Deklaration **Dim STELLE(1 To 8)** ein acht Zeichen langer Wert (Index) zugewiesen werden. Mit der Anweisung **BINAERZAHL = BINAERZAHL & STELLE(i)** erweitern Sie die Variable Binaerzahl um eine weitere Stelle.

**Hinweis:** Lösen Sie diesen Fall mit einer For...Next-Schleife (i = 1 to 8). Die Eingabe der Dezimalzahl geschieht über eine InputBox, die Ausgabe entsprechend über eine MessageBox.

### Notizen:

### 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.



### Exkurs Datentypumwandlungsfunktionen:

Die InputBox gibt die eingegebenen Werte als Text wieder, die für einen reibungslosen und sauberen Ablauf des Programmcodes in eine Zahl umgewandelt werden müssen.

#### VAL-Funktion

Gibt die in einer Zeichenfolge enthaltenen Zahlen als einen numerischen Wert eines geeigneten Typs zurück. So ermöglicht die Anweisung **ZAHLNEU = Val(ZAHL)**, das die Variable ZAHLNEU den numerischen Wert der eingelesenen Variable ZAHL erhält.

Für weitere Informationen recherchieren Sie bitte in der Online-Hilfe (*Typ-Umwandlungsfunktionen*)

**Hinweis:** Lösen Sie diesen Fall mit einer Do While ... Loop-Schleife. Die Eingabe der ganzen Zahl sowie des Modulaverfahrens geschieht über zwei InputBoxen, die Ausgabe entsprechend über eine MessageBox.

### Notizen: