****

**Schleifen in VBA – Info & Übung**

Schleifen ermöglichen es, dass Befehle mehrfach hintereinander ausgeführt werden, ohne dass dazu viel Programmcode erforderlich ist.

 **Code Was passiert hier?**

|  |  |
| --- | --- |
| Dim zeile, zaehler, ende as Integerzeile = 3ende = 10 | Variablen werden festgelegt (ganze Zahlen)Variable *zeile* wird auf 3 gesetzt, *ende* auf den Wert 10. |
|  |  |
| **For** zaehler = 0 To ende | Hier beginnt die Schleife. Alle Befehle zwischen **For** und **next** werden so lange ausgeführt, wie die Bedingung (zaehler ist nicht größer ist wie ende, d.h. 10) |
| Cells(*zeile*, 1).Value = Date + zahl | ***1. Durchlauf:*** *zeile* steht beim ersten Durchlauf auf 3 und zahl auf 0. Damit wird in die Zelle A3 das heutige Datum geschrieben.***2. Durchlauf:*** zeile steht auf 4; zahl auf 1. Damit wird in die Zelle A4 das morgige Datum geschrieben.***3. Durchlauf:*** zeile steht auf 5; zahl auf 2. Damit wird in die Zelle A4 das Datum von übermorgen geschrieben.(…) |
| Cells(*zeile*, 2).Value = zahl | Beim ersten Durchlauf wird in die Zelle B3 *0* geschrieben.Beim zweiten Durchlauf wird in die Zelle B4 1 geschrieben.(…) |
| Cells(*zeile*, 3).Value = zahl \* 2 | Beim ersten Durchlauf wird in die Zelle C3 *0* geschrieben.Beim zweiten Durchlauf wird in die Zelle C4 2 geschrieben.(…) |
|  |  |
| *zeile* = *zeile* + 1 | Die Variable *zeile* wird um 1 erhöht*.* |
| **Next** | Ende der Schleifen, zaehler wird um 1 hochgezählt. Die Schleife fängt wieder von vorne an, solange der zaehler <= 10 ist. Die Schleife wird somit 11-mal ausgeführt |

1. **Aufgabe:**Wie sieht das Tabellenblatt nach dem Ausführen dieses Makros aus? Überprüfe dies, indem Du ein solches Makro erstellst.

****

**Schleifen in VBA – Übung (2)**

Gegeben ist folgender Programmcode in VBA

|  |
| --- |
| **Dim zeile, spalte, ende, zahl As Integer** |
| **Cells(2, 2).Value = “Zahl1“** |
| **Cells(2, 3).Value = “Zahl2“** |
| **zeile = 3** |
| **spalte = 2** |
| **ende = 8** |
| **For zahl = 0 To ende** |
| **Cells(zeile, spalte).Value = zahl** |
| **Cells(zeile, spalte+1).Value = zahl\*2** |
| **zeile = zeile + 1** |
| **Next** |

**Aufgaben**

1. Überlege was durch Ausführen der einzelnen Programmzeilen passiert, und schreibe dies in die Tabelle oben neben die Befehle.
2. Trage die Ergebnisse dann handschriftlich in die Tabelle unten ein.
3. Überprüfe dann Dein Ergebnis, indem Du ein solches Makro erstellst.

****

**Schleifen – Übung 3**

Ergänze nun die Datei um ein weiteres Tabellenblatt mit dem Namen „Hintergrundfarben“.



Erstelle dann zwei Makros und zwei Schaltflächen:

***hintergrundfarbe()***

* Spalte B bekommt die Überschrift „Zahl“; Spalte C die Überschrift „Hintergrundfarbe“
* Es werden in Zahlen von 1 bis 56 in der Spalte B hochgezählt.
* In der Spalte C wird die entsprechende Hintergrundfarbe gesetzt.

**Schleifen in VBA – Übung (4)**

Schreibe nun ein Makro, das folgende Dinge ausführt.
***Tipp:*** Kopiere bei Bedarf schon vorhandene Befehle und passe sie dann an!

1. **Löschen** des Tabellenblattes.
2. ***Überschriften*** (s. Bild) in Grün, Fett, Schrifttyp Calibri und in bunt.
3. Es öffnet sich eine ***Inputbox*** (s. Bild)
4. Durch eine ***Schleife*** wird folgende Tabelle erzeugt. Dies Schleife ***endet***, wenn so viele Zeilen erzeugt worden sind, wie der Benutzer eingegeben hat.
	1. Datum: Datum von heute und in jeder Zeile wird es immer um einen Tag erhöht.
	2. Zahl 1: In der ersten Zeile steht 1 und es wird immer um 1 erhöht.
	Die Hintergrundfarbe entspricht dem Colorindex der Zahl
	3. Zahl 2: Zahl 1 \* 2.
	4. In Spalte D wird eine Zufallszahl zwischen 1 und 56 erzeugt und die Hintergrundfarbe entsprechend gesetzt.
5. Füge jetzt eine Umrandung hinzu (außen dick; innen dünn) und zentriere den Zelleninhalt, so dass Dein Tabellenblatt so aussieht:
6. Die Zufallszahlen sollen addiert werden und unter die Tabelle geschrieben sowie durch eine Messagebox ausgegeben.

**Info**

Eine **Zufallszahl** wird mit dem Befehl **Rnd** (random) erzeugt. Diese erzeugt eine Zufallszahl zwischen 0 und 1. Um eine Zahl zwischen 1 und 56 zu erhalten, muss daher diese Zahl noch mit 56 multipliziert werden. Da interiorcolorindex eine ganz Zahl erfordert, wird diese noch auf 0 Nachkommastellen gerundet.
Somit ergibt sich folgender Befehl: =***Round(Rnd \* 56, 0)***

**Umrandung**: Zeichne am besten ein Makro auf, das die Umrandung erstellt und *passe es dann an*. ***Beachte***: Die Umrandung hört bei der Tabelle auf. Dazu benötigst Du dann folgenden Befehl: ***Range(Cells(4, 1), Cells(Zeile - 1, 4)).Select***

Kommentiere die einzelnen Zeilen. Was bewirken die jeweiligen Befehle?

Range("C5").Font.Italic = True

Range("C5").Font.Bold = True

Range("C5").Font.Size = 24

Range("C5").Font.Name = Calibri

Range("C5").Font.Color = vbGreen

Range("B8").BorderAround LineStyle:=xlContinuous, Weight:=xlThin, Color:=vbBlack••••

Falls ein Rechner vorhanden, probiere die Makros aus!

**Tipp:** Befehle, die Du nicht kennst kannst Du gerne „googeln“!!

**Sub schleife1()**

Dim Zeile, Spalte, ende, zahl As Integer

Cells(2, 2).Value = "Zahl 1"

Cells(2, 3).Value = "Zahl 2"

Cells(2, 1).Value = "Date"

Cells(2, 4).Value = "Zufallszahl"

Zeile = 4

Spalte = 2

ende = 20

For zahl = 0 To ende

Cells(Zeile, Spalte).Value = zahl

Cells(Zeile, Spalte + 1).Value = zahl \* 2

Cells(Zeile, Spalte - 1).Value = Date + zahl

Cells(Zeile, Spalte + 2) = Round(Rnd \* 1000, 0)

Zeile = Zeile + 1

Next

End Sub

**Sub ShowDate()**

 MsgBox "Heute ist der " & Date

End Sub

**Sub ShowTime()**

Dim tag As String

'vor 12 Uhr

If Time < 0.5 Then tag = "Morgen"

If Time >= 0.5 And Time < 0.75 Then tag = "Tag"

If Time >= 0.75 Then tag = "Abend"

MsgBox "Guten " & tag & "! Wir haben jetzt " & Time & " Uhr."

End Sub

****

|  |  |  |  |
| --- | --- | --- | --- |
| Klasse | **Diff-Programmierung**VBA | 2024 | BKOLogo.jpg |
| W WFH12 | Kerstin Fröhlig |

**Bedingungen -1**

In Excel kennst Du schon die WENN-Funktion. In VBA (spricht ja nur Englisch) heißt dies **If-Funktion**. Ihre *Syntax* sieht wie folgt aus:

**If** *Bedingungsprüfung* **Then**

*Befehle, die dann ausgeführt werden*

**Else**
*Befehle, die sonst ausgeführt werden.*

**End If**

**Beispiel:**

If zahl1 < zahl2 Then

Cells(2, 3).Value = “Zahl1 ist kleiner als Zahl2“

Else

Cells(2, 3).Value = “Zahl1 ist nicht kleiner als Zahl2“

End If

**Aufgaben**

Programmiere drei Makros:

1. Loeschen
Es wird alles gelöscht.
2. ZahlenEingeben
-> Es werden die Überschriften gesetzt (Zahl1, …).
-> Durch Inputboxen gibt der Benutzer die Zahlen ein.
3. WerIstGroesser?
Bedingungsprüfung wie oben; Ist Zahl1 größer, dann soll die Hintergrundfarbe den Coloindex 34 haben, sonst 35.



|  |  |  |  |
| --- | --- | --- | --- |
| Klasse | **Diff-Programmierung**VBA | 2024 | BKOLogo.jpg |
| W WFH12 | Kerstin Fröhlig |

**Bedingungen -2**

So wie die Wenn-Funktion in Excel kann auch die If-Funktion verschachtelt werden, d.h. mehr als zwei Fälle berücksichtigen.

**If** *Bedingungsprüfung* **Then**

*Befehle, die dann ausgeführt werden*

**ElseIf** andere*Bedingungsprüfung**Befehle, die dann ausgeführt werden*

**<ElseIf ….>**

**Else**
*Befehle, die sonst ausgeführt werden.*

**End If**

**Aufgaben**

Erstelle ein neues Tabellenblatt (gleiche Datei wie Bedingung 1) und programmiere das folgende Makro.

**Tipp:** *Kopiere die entsprechenden Befehle aus dem AB Bedingung-1 und passe sie dann an.*

**WerIstGroesser2**

* Alle Zellen von A1 bis D6 werden gelöscht.
* Die Überschriften werden wie in AB1 gesetzt.
* Die Zellen für die Zahlen werden durch Zufallszahlen gefüllt.
* Prüfung: Ist die Zahl1 größer, kleiner oder gleich Zahl 2?

**Befehle:**

***Zufallszahl zwischen 0 und 100:*** *Round(Rnd \* 100, 0)*

**Hintergrundfarbe**: .Interior.ColorIndex = ..

***Eingabebox:***  Inputbox (“ Bitte …. „) )