

1 Základní konstrukce pro řízení toku

V této kapitole se podíváte na:

- ☆ Podmínkové konstrukce If, Iff a Case
- ☆ Smyčky, opakování neboli cykly s For a Do
- ☆ Zřetězení With
- ☆ Odskoky neboli větvení s GoTo

V programování se neobejdete bez několika zásadních věcí z nichž jednou jsou konstrukce pro řízení toků, mám tím na mysli, potřebu buď provádět činnosti pokud je splněna nějaká podmínka, případně provést požadovaný počet opakování, nebo odskočit na jinou část programu.

Proto se v této kapitole se podíváte na použití následujících konstrukce:

- If ...Then IfElse ... Else... End If
- Iff
- Select Case
- For ... Next
- Goto
- For ... Each
- Do Loop While | Until
- While ... Wend
- With

Materiály:



Materiály ke kapitole stahujte na: *v přípravě*

1.1 Podmínková konstrukce If ... ElseIf ... Else ... Then

Tato této podmínkové konstrukce se žádný programovací jazyk neobejde. Bez rozhodování se v programování využitím VBA neobejdete (stejně to máte v životě, tam se také denně rozhodujete). Když to zjednoduším prostě se rozhodnete a danou část kódu provedete nebo neprovedete, v programování je odpověď ano (True) nebo ne (False), jiná možnost prostě není, aby dotaz neskončil chybou.

1.1.1 Syntaxe

Syntaxe podmínkové funkce je jednoduchá:

```
If podmínka_1 Then
    výsledek_1
ElseIf podmínka_2 Then
    výsledek_2
ElseIf podmínka_n Then
    výsledek_n
Else
    výsledek_jinak
End If
```



V syntaxi je vidět, že If se může větvit s využitím Elseif, dle manuálu je v tomto případně vhodnější použít Select – Case což je zmíněno v další kapitole.

V případě požadavků můžete zjednodušit

```
If podmínka_1 Then
    výsledek_1
Else
    výsledek_jinak
End If
```

V případě ještě zjednodušit pro jednu podmínku, pokud chcete mít přehledně

```
If podmínka_1 Then
    výsledek_1
End If
```

Nebo ještě na jeden řádek

```
If podmínka 1 Then výsledek 1
```

Tip existuje ještě funkce If, kdy na jeden řádek vměstnáte podmínku pro ano i ne. O této funkci viz další kapitola.

1.1.1.1 Operátory

Pro zopakování, si jen připomeňme, že v podmínkách můžete použít porovnávací operátory (=,>,<,...), logické operátory (And, Or, ...) speciální operátory (Like a Is). Viz samostatná kapitola týkající se **operátoru**.

<	And	Is
>	Or	Like
>=	Not	
<=	Xor	
<>		
=		

1.1.2 Praktické ukázky If Then

Základní teorií máte za sebou, tak se můžete vrhnout na praktické příklady

1.1.2.1 Odpověď ano/ne na dotaz v dialogovém okně (MsgBox)


Začneme jednoduchou variantou kdy máte jako odpověď pouze dvě hodnoty (ano,ne), neboli (yes,no), neboli (Pravda,Nepravda) v angličtině (True,False).

V dialogovém okně se zeptáte: „Jsi muž?“ pokud člověk klikne na Ano (Yes) tak se zobrazí dialogové okno: „Jsi může.“ V opačném případě, kdy klikne na Ne (No) tak se zobrazí odpověď: „Jsi žena“.

V ukázce je využíváno dialogové okno MsgBox, které je popsáno v samostatné kapitole **Dialogové okno MsgBox**.


Poznámka: Předpokládám, že v této ukázce existují pouze muži a ženy!

```
i = MsgBox("Jsi muž?", vbYesNo, "Kdo jsi")
'i = 6 'Nechcete-li využívat dialogové okno, vložte do proměnné i hodnotu
'druhá návratová hodnota z MsgBox vbYes potažmo vbNo je popsána dále
If i = 6 Then
    MsgBox ("Jsi muž.")
End If
If i = 7 Then
    MsgBox ("Jsi žena.")
End If
```



Nevýhoda že máte dvě syntaxe If – Then – End If, někoho napadne využít jeden řádek

```
'i = MsgBox("Jsi muž?", vbYesNo, "Kdo jsi")
If i = 6 Then MsgBox ("Jsi muž.")
If i = 7 Then MsgBox ("Jsi žena.")
```



Což v ukázce super, ale v praxi kdy máte více příkazu, tak toto není ideální řešení, můžete využít a znalosti, že máme pouze muže a ženy a využít v syntaxi klíčové slovo Else:

```
i = MsgBox("Jsi muž?", vbYesNo, "Kdo jsi")
If i = 6 Then
    MsgBox ("Jsi muž.")
Else
    MsgBox ("Jsi žena.")
End If
```

Případně pokud chcete místo hodnot, které vrací tlačítko v MsgBoxu se odkázat na konstanty, které se vrací současně při stisku vybraného tlačítka, můžete použít konstanty vbYes a vbNo

```
i = MsgBox("Jsi muž?", vbYesNo, "Kdo jsi")
If i = vbYes Then
    MsgBox ("Jsi muž.")
End If
If i = vbNo Then
    MsgBox ("Jsi žena.")
End If
```

Případně zkrátit s využitím Else

```
i = MsgBox("Jsi muž?", vbYesNo, "Kdo jsi")
If i = vbYes Then
    MsgBox ("Jsi muž.")
Else
    MsgBox ("Jsi žena.")
End If
```

1.1.2.2 Větvení na základě více podmínek - věk

Chcete rozhodnout podle věku, který se zadává do InputBoxu (více o Input box v kapitole **InputBox**). Kdy chcete rozhodnout na základě pravidel:

- Pokud je pod 15 let, tak je dítě – neboli vypsát „Jsi dítě.“
- Pokud pod 30 a větší rovno 15 tak vypsát „Jsi mladoch.“
- Pod 60 a větší rovno 30 vypsát „Jsi v nejlepších letech.“
- Od 60 včetně a více vypsát „Jsi v důchodu.“

```
Vek = InputBox("Zadej svůj věk:", "Ukázka then více", 0)
If Vek < 15 Then
    MsgBox "Jsi dítě."
ElseIf Vek < 30 Then
    MsgBox "Jsi mladoch."
ElseIf Vek < 60 Then
    MsgBox "Jsi v nejlepších letech."
Else
    MsgBox "Jsi v důchodu."
End If
```

Tip: Lze řešit i přes Select Case viz kapitola **Case Select**.

1.1.2.3 Provedení po splnění všech nebo jedné a více podmínek (OR – AND)

Potřebuji vypsát „Splňuje.“ Pokud je město Ostrava zadáno do proměnné Mesto jako zkratka „OV“ a věk je větší jak 25, zadáno do proměnné Vek jako číslo.

Opět z důvodu zjednodušení není provedena deklarace proměnných a proměnné jsou zadány přímo v kódu.

```
Mesto = "OV"
Vek = 35
If Mesto = "OV" And Vek > 25 Then
    MsgBox ("Splňuje.")
End If
```

Druhá možnost je potřebujete mít splněno pokud bude město z Ostravy nebo věk větší jak 25 let.

```
Mesto = "OV"
Vek = 35
If Mesto = "OV" Or Vek > 25 Then
```

```
MsgBox ("Splňuje.")
End If
```

1.1.2.4 Více podmínek a možné komplikace - závorky

V této kapitole si ukážeme možné problémy pokud je podmínek více v kombinace, neboli něco musí být splněno, něco buď jedno nebo druhé, nebo-li kombinace and a or. V ukázce potřebuji vypsat „Splňuje.“ Pokud je město Ostrava zadáno do proměnné Mesto jako zkratka „OV“ nebo Brno zadáváno jako zkratka „BR“, ale věk je větší jak 25, zadáno do proměnné Vek jako číslo. Splní pokud bude člověk z Brna mít větší věk než 25, stejně tak člověk z Ostravy pokud bude mít věk větší jak 25 let.

```
' Chybné řešení
Mesto = "OV"
Vek = 20
' Chybí závorky vyhodnotí chybně jako Splněno
If Mesto = "OV" Or Mesto = "Brno" And Vek > 25 Then
    MsgBox ("Splnuje.") ' ač věk pod 25!!
End If
```

Vhodnější řešení za využití závorek

```
' Možné řešení se závorkami
Mesto = "OV"
Vek = 35

If (Mesto = "OV" Or Mesto = "Brno") And Vek > 25 Then
    MsgBox ("Splňuje")
Else
    MsgBox ("Neplnuje")
End If
```

Tip: Můžete si představit logický operátor **or** jako sčítání a logický operátor **and** jako násobení. Pouze jako ukázka:

```
Mesto = "OV"
Vek = 20
If (Mesto = "OV" Or Mesto = "Brno") And Vek > 25 Then
    MsgBox ((Mesto = "OV" + Mesto = "Brno") * Vek > 25)
    MsgBox "Splňuje."
Else
    MsgBox ((Mesto = "OV" + Mesto = "Brno") * Vek > 25)
    MsgBox "Neplňuje."
End If
```

Ukázka pro chybné řešení, viz kapitola **práce s operátory**.

```
' Chybné řešení
Mesto = "OV"
Vek = 20

If (Mesto = "OV") + (Mesto = "Brno") * (Vek > 25) Then
    MsgBox ((Mesto = "OV") + (Mesto = "Brno") * (Vek > 25))
    MsgBox (True + False * False)
    MsgBox ("Splňuje ač věk pod 25!!")
Else
    MsgBox (Mesto = "OV" + Mesto = "Brno" * Vek > 25)
    MsgBox ("Splnuje ač věk 25!!")
End If
```

1.1.2.5 Dvě podmínky a ošetřit všechny možnosti

Potřeba ošetřit dvě zadávané podmínky, v první odpovědi proměnná *Odpoved1* se zadává pohlaví (F pro ženu nebo M pro muže) a v druhé proměnné *Odpoved2* se zadává oddělení (PR – z propagace a HR z náborového). Potřebujete na základě odpovědi určit:

- Jsi žena z oddělení HR.
- Jsi žena z oddělení PR.
- Jsi muž z HR.
- Jsi muž z PR.

Asi vás napadne vytvořit kód:

```

Odpoved1 = InputBox("Zadej pohlaví:", "Pohlaví", 0)
Odpoved2 = InputBox("Zadej oddělení:", "Oddělení", 0)

If Odpoved1 = "F" And Odpoved2 = "HR" Then
    MsgBox ("Jsi žena z oddělení HR.")
ElseIf Odpoved1 = "F" And Odpoved2 = "PR" Then
    MsgBox "Jsi žena z oddělení PR."
ElseIf Odpoved1 = "M" And Odpoved2 = "HR" Then
    MsgBox "Jsi muž z HR."
Else
    MsgBox "Jsi muž z PR."
End If

```

Pokud je možno zadat jen požadované parametry bude kód fungovat. Musíte ale někdy brát úvahu, že při zadávání můžete mít překlep a místo M zadat Ž (*jako žena*), pak by tato osoba byla zařazena do sekce „Jsi muž z PR.“ Což by bylo špatně. Kód můžete vylepšit o další podmínku, kdy v případě chyby bude zobrazeno:

- „Chybně zadané parametry nelze určit.“

```

Odpoved1 = InputBox("Zadej pohlaví:", "Pohlaví", 0)
Odpoved2 = InputBox("Zadej oddělení:", "Oddělení", 0)

If Odpoved1 = "F" And Odpoved2 = "HR" Then
    MsgBox "Jsi žena z oddělení HR."
ElseIf Odpoved1 = "F" And Odpoved2 = "PR" Then
    MsgBox "Jsi žena z oddělení PR."
ElseIf Odpoved1 = "M" And Odpoved2 = "HR" Then
    MsgBox "Jsi muž z HR."
ElseIf Odpoved1 = "M" And Odpoved2 = "PR" Then
    MsgBox "Jsi muž z PR."
Else
    MsgBox "Chybně zadané parametry nelze určit."
End If

```

1.1.2.6 Vnořování podmínek

Jako v předchozí ukázce potřebujete ošetřit dvě zadávané podmínky, v první proměnná *Odpoved1* zadáváte pohlaví (F pro ženu nebo M pro muže) a v druhé *Odpoved2* zadáváte oddělení (PR – z propagace a HR z náborového). Potřebujete na základě odpovědi určit:

- Jsi žena z oddělení HR.
- Jsi žena z oddělení PR.
- Jsi muž z HR.
- Jsi muž z PR.

V řešení využijete vnoření podmínky do podmínky

```

i ' Předpoklad v odpovědi nelze mít překlep
Odpoved1 = InputBox("Zadej pohlaví:", "Pohlaví", 0)
Odpoved2 = InputBox("Zadej oddělení:", "Oddělení", 0)
If Odpoved1 = "F" Then
    If Odpoved2 = "HR" Then
        MsgBox ("Jsi žena z HR.")
    Else
        MsgBox ("Jsi žena z PR.")
    End If
Else
    If Odpoved2 = "HR" Then

```

```

        MsgBox ("Jsi muž z HR.")
    Else
        MsgBox ("Jsi muž z PR.")
    End If
End If

```

Poznámka: Co když může dojít k chybě při zadávání? Vylepšíte již určitě sami.

1.1.2.7 Ošetření chyb

Ošetření chyb je popsáno v samostatné kapitole **jak na ošetření chyb** ve VBA, ze jen ukázka pro doplnění.

```

Dim number As Integer
On Error GoTo nalezen_problem
number = InputBox("Zadej číslo: ")
If number Mod 2 = 0 Then
    MsgBox ("Číslo sude!")
Else
    MsgBox "Číslo liché!"
End If
Exit Sub
nalezen_problem:
MsgBox ("Problém, asi nebylo zadáno číslo :)")

```

1.2 Podmínková konstrukce – funkce IIf

Jak na podmínkovou funkci IIf v případě kdybyste na touto situaci narazily. Jako další možnost, která je málo využívána.

1.2.1 Syntaxe IIf

Syntaxe funkce IIf:

```
IIf (podmínka, pokud_pravdiva, pokud_nepravdiva)
```

1.2.2 Praktická ukázka použití IIf

Opět využijete dialogové okno s dotazem na velikost platu. Pokud uživatel odpoví (klikne na Ano) zobrazí se dialogové okno „Odpověď ANO.“ Pokud klikne na ne tak se zobrazí dialogové okno „Odpověď je Ne.“. Tento úkol určitě hravě vyřešíte s využitím If Then Else. V této ukázce použijeme funkci IIf (připomínající z Excel funkce KDYŽ/IF). Na jeden řádek kódu vměstnáte obě odpovědi s využitím MsgBox.

```

i = MsgBox("Chcete větší plat?", vbYesNo, "Plat")
MsgBox (IIf(i = vbYes, "Odpověď ANO.", "Odpověď NE.))

```

1.3 Podmínková konstrukce Select Case

Další možností pro větvení programů je využití konstrukce Select Case.

1.3.1 Syntaxe Select - Case

```

Select Case testovaný_výraz
    [Case seznam_výrazů-n [příkazy-n]]
[Case Else
    [elsepříkazy]]
End Select

```

1.3.1.1 Operátory

V podmínkách můžete použít porovnávací operátory (=,>,<..), speciální operátory (Like a Is) a kontrolované hodnoty oddělovat čárkou. Viz samostatná kapitola týkající se **operátoru**.

<	Is
>	Like

>=	To
<=	
<>	
=	

1.3.2 Praktické ukázky použití Select Case

V následujících podkapitolách se podíváte na možná řešení využitím Select Case

1.3.2.1 Kontrola dvou hodnot

Potřebujete zkontrolovat zda čísla zadávaná do dvou InputBoxu jsou stejné:

```
a = InputBox("Zadej číslo 1:")
b = InputBox("Zadej číslo 2:")
Select Case a = b
Case True
  MsgBox ("Zadány dvě stejné hodnoty.")
Case False
  MsgBox ("Zadány různé hodnoty.")
End Select
```

Poznámka: Řešení využitím If else

Tip: Co se stane pokud zadáte písmena? Náповěda nejsou deklarované proměnné.

1.3.2.2 Příklad na rozsahy - známkování

V ukázce potřebujete na základě zadaných bodů rozhodnout o známce, body mohou být pouze číslo v rozsahu 0 až 100 (proto dále nemusíte ošetřovat a kontrolovat):

- Rovno a větší 90 – text „A – velice super.“
- Rovno a větší 80 – text „B - super“
- Rovno a větší 60 – Text „C - nic moc“
- Pro všechny ostatní případy „E - špatné“

```
' ukázka deklarace proměnných
Dim body As Integer
Dim vysledek As String
' Načtení hodnoty z buňky
body = Range("B10").Value
' Můžete využít i input box nebo zadat do proměnné přímo
Select Case body
  Case Is >= 90
    vysledek = "A - velice super"
  Case Is >= 80
    vysledek = "B - super"
  Case Is >= 60
    vysledek = "C - nic moc"
  Case Else
    vysledek = "E - špatné"
End Select
```

1.3.2.3 Příklad na rozsahy od do – (Is)

Potřebujete pokud je zadané číslo do proměnné *Hodnota* v rozsahu větší rovno 8 a menší rovno 12 vypsat text „Od 8 do 12“ pokud ne tak text „Ostatní“.

```
' Otestujte změnu čísla v proměnné a ne jen číslo ;)
Hodnota = 10

Select Case Hodnota
Case Is >= 8, Is <= 12
  MsgBox ("Od 8 do 12")
Case Else
```

```
MsgBox ("Ostatní")
End Select
```

1.3.2.4 Výčet hodnot – využitím čárky

Elegantně jde v Select case využít výčet hodnot, které stačí oddělovat čárkou. Neboli do proměnné *Hodnota* zadáváte využitím InputBoxu názvy ovoce a zeleniny a pokud zadaný název je zelenina, nebo ovoce ve výčtu do dialogového okna vypíšete:

- „Jde o ovoce ...“
- „Jde o zeleninu ...“
- „Nemůžu rozhodnout.“

```
i Hodnota = InputBox("Zadej ovoce, zelenina:")
' rozhodnutí co bylo zadáno - ovoce, zelenina
Select Case Hodnota)
' přidejte další ovoce a zeleninu a otestujte
Case "Jablko", "Hruška", "Banán", "Jahoda", "Jablko"
  MsgBox ("Jde o ovoce ...")
Case "Mrkev", "Petržel", "zelí"
  MsgBox ("Jde o zeleninu ...")
Case Else
  MsgBox ("Nemůžu rozhodnout.")
End Select
```

1.3.2.5 Praktická ukázka rozsah a obsah textu

Využitím To můžete testovat rozsah zadávaných slovních spojení. Viz kapitola **Like a is podrobněji**.

```
' Ukázka pro rozsah textu
Hodnota = InputBox("Zadej text:")
Select Case Hodnota
  Case "aa" To "dd"
    MsgBox ("Text: aa - dd")
  Case Else
    MsgBox ("Ostatní text")
End Select
```

Podobně můžete využít slovo Like, texty zadáváte do proměnné slovo

```
slovo = "JakNaExcel"
Select Case True
'pokud text obsahuje písmena Na
  Case slovo Like "*Na*"
    MsgBox ("Nalezeno")
  Case Else
    MsgBox ("Nenelezeno")
End Select
```

1.3.2.6 Ukázka reakce na tlačítka v dialogovém okně

Pokud chcete podobně jako u konstrukce if reagovat na stisky tlačítek. Více v článku o **MsgBox** a v **deklaraci proměnných**.

```
'' Ukázka dialogového boxu
'' Viz deklarace proměnných
Dim Odpoved As VbMsgBoxResult

Odpoved = MsgBox("Něco stiskni!", vbYesNoCancel)

Select Case Odpoved
  Case vbYes
    MsgBox ("Klikl jsi na Ano - Yes")
  Case vbNo
    MsgBox ("Klikl jsi na Nbe - No")
  Case vbCancel
```



```
MsgBox ("Klikl jsi na Zrušit - Cancel")  
End Select
```

1.4 Cyklus For next

1.4.1 Syntaxe

```
For čítač = začátek To konec [Step krok]  
    [příkazy]  
Next [čítač]
```

1.4.1.1 Násilné Ukončení programu

Při testování mnohdy může dojít k zacyklení programu, běh programu můžete ukončit (*doporučuji otestovat, ať v případě vážné komplikace víte jak a nemusíte provést restart Excel a přijít o svůj kus kódu*):

- Přerušení programu: Ctrl + Break
- Pozor notebooky třeba: Ctrl + Fn + B

1.4.2 Praktické použití cyklů For next

1.4.2.1 Základní cyklus od do

Potřebuji vypsat čísla od jedné do deseti do dialogového boxu

```
For i = 1 To 10  
    MsgBox (i)  
Next i
```

Potřebuji vypsat čísla od deseti do jedné do dialogového boxu

```
For i = 10 To 1 Step - 1  
    MsgBox (i)  
Next i
```

1.4.2.2 Zápis čísel do buněk

Potřebujete zapsat čísla 1 až 10 do buněk ve sloupci A, od řádku 1 do řádku 10. Pro ukázkou využíváme Cells, pro dostudování této problematiky viz kapitola **zápis do buněk**.

```
Dim i As Integer  
For i = 1 To 10  
    Cells(i, 1).Value = i  
Next i
```

1.4.2.3 Násilné ukončení cyklu využitím - Exit For

Potřebujete vypisovat čísla od jedničky do desítky, ale při čísle osm potřebujete z cyklus ukončit. Záměrně využitím Exit For:

```
Dim i As Integer  
For i = 1 To 10  
    Cells(i, 4).Value = i  
    If i = 8 Then  
        Exit For  
    End If  
Next i
```

1.4.2.4 Vnoření cyklu – dvojrozměrné pole

Úkolem bude vyplnit buňky od řádku 1 do řádku 10 a od sloupce 1 (neboli A) do sloupce 10 (neboli J) malou násobilkou, kdy hodnota na průsečíku řádku a sloupce bude jeho číslo. K zápisu používám kód Cells. Více informací v kapitole **jak zapsat hodnoty do buňky**.

```
Dim i As Integer  
Dim j As Integer
```

```
For i = 1 To 10
  For j = 1 To 10
    Cells(i, j).Value = i * j
  Next j
Next i
```

1.4.2.5 Kombinace cyklu a podmínek

Nemusíte vnořovat do sebe jen podmínky a cykly, můžete kombinovat do cyklu vnořit podmínku a opačně. V ukázce budu potřebovat zobrazovat čísla pokud jsou větší než 5, ale procházet budete v cyklu čísla od jedničky.

```
For i = 1 To 10
  IF i > 5 Then
    MsgBox (i)
  End If
Next i
```

1.4.2.6 Zápis do oblastí

Úkolem bude zapsat do buněk od řádku 5 do řádku 8 a od sloupce 5 do sloupce 8 hodnotu 100. Ukázka je záměrně z důvodu porovnání lepšího řešení využitím For Each, vid další kapitola **For Each**.

```
For i = 5 To 8
  For j = 5 To 8
    Cells(i, j).Value = 100
  Next j
Next i
```

1.4.2.7 Vyplnit náhodná čísla ve matici buněk (sloupci, řádku)

Potřebujete zapsat náhodná čísla do oblasti. K zápisu využijete **Rnd**, lepším řešením mnohy bude využít **For Each**.

```
Dim Sloupec As Long
Dim Radek As Long
For Sloupec = 1 To 10
  For Radek = 1 To 10
    Cells(Radek, Sloupec) = Rnd
  Next Radek
Next Sloupec
```

1.4.2.8 Podbarvit každý třetí řádek

Dalším úkolem kde využijete cyklus je grafický podoba tabulek, kdy například potřebujete podbarvovat řádky, například každý třetí. Více o práci s barvou pozadí řádku v kapitole – **barva textu a pozadí buňky**. V první ukázce potřebujete pro prvních 100 řádku.

```
Dim i As Long
For i = 1 To 100 Step 3
  Rows(i).Interior.Color = RGB(200, 200, 200)
Next i
```

Ve druhé ukázce vylepšíte a na základě **zjištění posledního řádku** vylepšíte kód:

```
Dim PosledniPlnyRadek As Long
PosledniPlnyRadek = Cells(Rows.Count, "A").End(xlUp).Row ' Ve sloupci A

For PosledniPlnyRadek = 1 To 100 Step 3
  Rows(i).Interior.Color = RGB(200, 200, 200)
Next PosledniPlnyRadek
```

1.4.2.9 Šachovnice

Když už jsme u toho zabarvování můžete vytvořit matici (šachovnici), opět využitím vnořeného příkazu For:

```
Dim R As Long, C As Long
For R = 1 To 8
  If WorksheetFunction.IsOdd(R) Then
    For C = 1 To 8 Step 2
      Cells(R, C).Interior.Color = 255
    Next C
  Else
    For C = 2 To 8 Step 2
      Cells(R, C).Interior.Color = 255
    Next C
  End If
Next R
```

1.4.2.10 Úkol je potřeba mít od zadaného měsíce mínus 3 měsíce

Dalším zajímavým úkolem je od zadaného měsíce odečíst tři. Problém nastane pokud jde o leden. V případě klasického odečtení máme měsíce 0, -1, -2 což nechcete, potřebujete měsíce 12,11,10. A samozřejmě chcete i předchozí rok!

Kód je doplněn i o ukázkou pokud potřebujete mít měsíce jako dvouciferné, pro březen mít 03 a ne pouze číslo 3.

```
' Můžete měnit proměnné Y kde je rok a M kde je měsíc a
' v cyklu kolik měsíců chcete odečíst

Y = 2020
M = 12
For i = 1 To 3 ' počet měsíců zde 3
  M = M - 1
  If M = 0 Then
    M = 12 - M
    Y = Y - 1 ' rok
  End If
  ' Pokud potřebuji dvouciferné měsíce tj 01, 02
  If M < 10 Then
    'M = "0" & M ' pokud měsíc < 10 doplnit "0"
  End If
  MsgBox ("Mesic: " & M & " Rok: " & Y)
Next i
```

1.4.2.11 Úkol je potřeba mít od zadaného měsíce plus 3 měsíce

Když už umíme požadovaný počet měsíců odečíst hodí se umět i požadovaný počet měsíců přičíst:

```
' Můžete měnit Y rok a M kde je měsíc
Y = 2020
M = 12
For i = 1 To 3 ' počet měsíců zde 3
  M = M + 1
  If M > 12 Then
    M = M - 12
    Y = Y + 1 ' rok
  End If
  ' Pokud potřebuji dvouciferné měsíce tj 01, 02
  If M < 10 Then
    'M = "0" & M ' pokud mesic < 10 doplnit "0"

  End If
  MsgBox ("Mesic: " & M & " Rok: " & Y)
Next i
```

1.4.2.12 Hody kostkou

Dalším pěkným příkladem na použití vnořených cyklu je výpis všech hodů třemi kostkami. Klidně si můžete rozšířit na šest kostek ;). Jen pozor u **deklarace proměnné** i půjde o Long. Hodnoty vypisujte do buněk. Kdyby náhodou někoho napadlo vypsat do dialogového boxu tak se hodí umět přerušit běh programu, kombinací je opravdu hodně (množství větší než malé).

```
Dim a As Integer
Dim b As Integer
Dim c As Integer
Dim i As Integer
i = 2
For a = 1 To 6
    For b = 1 To 6
        For c = 1 To 6
            Cells(i, 1).Value = a
            Cells(i, 2).Value = b
            Cells(i, 3).Value = c
            Cells(i, 7).Value = a + b + c
            i = i + 1
        Next c
    Next b
Next a
```

1.4.2.13 Výpis hodnot v pole (Array)

Cykly se dají s výhodou použít i u **práce s poli** (v angličtině Array). Jako ukázky vypsaní hodnot v poli.

```
i pole = Array("Eva", "Jan", "Iva")

For i = LBound(pole) To UBound(pole)
    MsgBox (pole(i))
Next i
```

1.5 Odskoky – větvení programu využitím Goto

V Excel VBA můžete využívat i odskoky. Nevýhodou je komplikovaná kontrola a testování kódu, proto pouze jako zmínka použití.

1.5.1 Praktická ukázka použití GoTo

Prakticky na „odskoky“

1.5.1.1 Odskoky na demonstraci možných problémů

V ukázce potřebujete vyhodnotit zadané číslo a podle zadaného čísla se následně přesunete na příslušný štítek:

- GoTo Radek1
- GoTo Radek2

Pokud činnosti v dané sekci Radek1, případně Radek2 dokončíte, potřebujete se přesunout na konec využitím odskoku na štítek PosledniRadek:

- GoTo PosledniRadek

V případě, že někde odskok zapomenete může se začít provádět operace z následujícího odskoku a program nebude fungovat, jak potřebujete. V zapomenutí definice kam směřovat, jsou odskoky velice zrádné.

V ukázce si „zapoznámkujete“ GoTo PosledniRadek sledujte co se bude dít v případě změny v parametru Cislo = 1 číslo z 1 na 2.

```
' Deklarace
Dim Cislo, MujRetezec
' Inicializuj proměnnou.
' testovat přes inputBox
Cislo = 1

' Vyhodnoť Cislo a pro větvení vyber příslušné návěští.
If Cislo = 1 Then
    GoTo Radek1
Else
    GoTo Radek2
End If

Radek1:
    MujRetezec = "Číslo se rovná 1"
    ' Jdi na PosledniRadek pokud nebude uvedeno
    ' provede se i následující krok bez ohledu na to že je uvozen štítkem
    GoTo PosledniRadek
    ' může se ukončit Exit Sub

Radek2:
    ' Následující příkaz se nikdy neprovede.
    MujRetezec = "Číslo se rovná 2"
    GoTo PosledniRadek

PosledniRadek:
    MsgBox (MujRetezec)
```

1.5.1.2 Práce s chybami

S velkou výhodou jde GoTo použít při „odchytávání“ chyb. V ukázce kontrolujete za je číslo sude nebo liché a využíváte k tomu zadání z InputBox. Pokud ale uživatel zadá text u dělení number Mod 2 vznikne chyba. Pokud ji ale ošetříte On Error GoTo nalezen_problem můžete tuto chybu nasmělovat na příslušný štítek nalezen_problem:.

```
Dim number As Integer
On Error GoTo nalezen_problem
number = InputBox("Zadej číslo: ")
If number Mod 2 = 0 Then
    MsgBox ("Číslo sude!")
Else
    MsgBox "Číslo liché!"
End If
Exit Sub
nalezen_problem:
MsgBox ("Problém, asi nebylo zadáno číslo :)")
```

1.6 For ... Each

Pokud je potřeba opakovat nad všemi prvky kolekcemi. Syntaxe může připomínat konstrukci For Next

1.6.1 Syntaxe

```
For Each element In group
    [ statements ]
    [ Exit For ]
    [ statements ]
Next [element]
```

1.6.2 Prakticky

Pojďme se podívat na praktické použití

1.6.2.1 Výpis všech listů v sešitě

Úkolem bude vypsat název listu v sešitě. Pokud víte kolik je v sešitě listů můžete řešit i využitím For – Next, ale elegantnější je využít For Each, neboli pro všechny objekty v tomto případě pro všechny list proved' a to jejich vypsání názvu daného listu do dialogového okna. Více **o práci s listy** v samostatné kapitole.

```
Dim ws As Worksheet
' ws definovano jako list
For Each ws In ActiveWorkbook.Worksheets
    MsgBox (ws.Name)
Next ws
```

1.6.2.2 Zavřít všechny otevřené sešity

Potřebujete-li zavřít všechny otevřené sešity

```
Dim wb As Workbook

For Each wb In Workbooks
    wb.Close SaveChanges:=True
Next wb
```

1.6.2.3 Smazat všechny objekty v aktivním listě

Jak smazat všechny objekty (obrázky, nákresy...) v aktivním sešitě.

```
Dim shp As Shape

For Each shp In ActiveSheet.Shapes
    shp.Delete
Next shp
```

1.6.2.4 Výpis hodnot buněk v oblasti

Budete-li chtít vypisovat hodnotu buněk v oblasti, v ukázce do dialogového okna (viz **kapitola jak přechíst hodnotu z buňky**), případně můžete v testech vypisovat hodnoty do **Debug.Print**.

```
Dim rng As Range
Dim cel As Range

Set rng = Range("B4:C5")

' Pro všechny buňky v oblasti
For Each cel In rng.Cells
    MsgBox (cel.Value)
    ' mohu vypisovat více informací
    ' With cel
    '     Debug.Print .Address & ":" & .Value
    ' End With
Next cel
```

1.6.2.5 Násilné ukončení cyklu For Each využitím Exit For

Stejně jako o For Next můžete v konstrukci For Each použít příkaz Exit pro „násilné“ ukončení provádění. V ukázce budete vypisovat názvy listů a v případě, že v sešitě více jak 4 listy tak u 4 listu dojde k ukončení a názvy dalších list se nezobrazí.

```
k = 1
For Each List In ActiveWorkbook.Worksheets
    k = k + 1
    MsgBox List.Name
    If k = 4 Then Exit For
Next List
```

1.6.2.6 Vyplnění oblastí pro porovnání s řešením For Next

Pro porovnání s předchozí ukázkou v kapitole For next, kdy potřebujete vyplnit oblast, kdy znáte které řádky a sloupce potřebujete vyplnit:

```
radekZac = 5
radekKon = 7
sloupZac = 5
sloupKon = 8
For Each cell In ActiveSheet.Range(Cells(radekZac, sloupZac),
Cells(radekKon, sloupKon))
    cell.Value = 50
Next cell
```

1.7 Do Loop Until

1.7.1 Syntaxe

Existují dva druhy smyčky:

- While - dokud je podmínka splněna
- Until - dokud podmínka není splněna

Navíc v obou podmínkách může být kontrola

- Na začátku
- Na konci

Což bude mít vliv na výsledek. Navíc oproti For Next, kde skoky mohli být pouze celá čísla o konstrukce Do může jít i o desetinná čísla. Krásné využití nalezne v případech kdy počet opakování není znám, ale záleží na nějaké podmínce.

1.7.1.1 Syntaxe s vybranou podmínkou na začátku

```
Do [{While | Until} podmínka]
    [příkazy]
    [Exit Do]
    [příkazy]
Loop
```

1.7.1.2 Syntaxe s vybranou podmínkou na konci

```
Do
    [příkazy]
    [Exit Do]
    [příkazy]
Loop [{While | Until} podmínka]
```

Argumenty:

- **podmínka** - volitelné. Číselný výraz nebo řetězcový výraz, který je vyhodnocen jako True nebo False. Je-li podmínka Null, pak je vyhodnocena jako False.
- **příkazy** - Jeden nebo více příkazů, které jsou opakovány, dokud je nebo dokud není podmínka True.

1.7.1.3 Pouze Do Loop s podmínkou

Lze využít i případ Do – Loop bez uvedení podmínky, ale musí obsahovat příkaz na Exit, jinak se program zacyklí. Proto stejně jako o For Next platí umět z programu „vyskočit“ umět program natvrdo přerušit (Ctrl + Break).

Další odlišností je že počítadlo se nepočítá samo, musíte tedy počítat externě. Navíc pozor při práci s desetinnými čísly v počítadle. Díky desítkové soustavě může dojít k „zaokrouhlení a 0,1 nemusí být 0,1, ale 0,09999999998, což následně vám může zacyklit program.

1.7.1.4 Ekvivalent WhileWend

Ekvivalent Do While – Loop.

```
While podmínka  
    [příkazy]  
Wend
```

Argumenty:

- podmínka Povinné. Numerický výraz nebo řetězcový výraz, který je vyhodnocen jako True nebo False.
- příkazy Volitelné. Jeden nebo více příkazů provedených, dokud je podmínka True.

1.7.2 Do Loop – prakticky

Pojďte se podívat na praktické příklady, kde pochopíte jak konstrukce Do Loop pracují.

1.7.2.1 Počítadlo Do While s podmínkou na začátku

Potřebuji do buněk zapsat čísla 1 až 10.

```
i Pocitadlo = 1  
Do While Pocitadlo <= 10 ' Vnitřní smyčka.  
    ' dokud je pravda prováděj  
    Cells(Pocitadlo + 4, 1) = Pocitadlo  
    Pocitadlo = Pocitadlo + 1 ' Zvyš počítadlo.  
Loop ' Dopočítá až 10
```

1.7.2.2 Počítadlo Do While s podmínkou na konci

Stejně jako v předchozím případě potřebujete dopočítat do 10, ale podmínku uvedete na konci. Tím pádem vám konstrukce Do Loop dopočítá jen do 9. Neboli tento příkaz se vykoná alespoň jednou.

```
i Pocitadlo = 1  
Do  
    Cells(Pocitadlo + 4, 2) = Pocitadlo  
    Pocitadlo = Pocitadlo + 1 ' Zvyš počítadlo.  
    ' zapiše a pak se teprve zkontroluje  
    ' neboli zapiše 10 a pak zkontroluje parametr  
Loop While Pocitadlo <= 10 ' Vnitřní smyčka. ' Dopočítá jen do 9
```

1.7.2.3 Počítadlo Do Until s podmínkou na začátku

Příklad s Do Until s podmínkou na začátku

```
Pocitadlo = 1  
Do Until Pocitadlo >= 10 ' Vnitřní smyčka.  
    Cells(Pocitadlo + 4, 1) = Pocitadlo  
    Pocitadlo = Pocitadlo + 1 ' Zvyš počítadlo.  
    ' MsgBox (pocitadlo)  
Loop
```

1.7.2.4 Počítadlo Do Until s podmínkou na konci

Příklad s Do Until s podmínkou na konci

```
Pocitadlo = 1  
Do  
    Cells(Pocitadlo + 4, 2) = Pocitadlo  
    Pocitadlo = Pocitadlo + 1 ' Zvyš počítadlo.  
Loop Until Pocitadlo >= 10 ' Vnitřní smyčka.
```

1.7.2.5 Zápis do buněk text a číslo

Do buněk potřebujete zapsat statický text Excel a pořadové číslo.


```
i Dim n As Integer
n = 0
Do While n < 11
    n = n + 1
    Range("A" & n).Value = "Excel " & n
Loop
```

1.7.2.6 Do Loop s „násilným“ ukončením

Potřebujete zapsat čísla do buněk, od 1 do 5 s využitím konstrukce Do bez podmínky s „násilným“ ukončením pomocí Exit Do.

```
Pocitadlo = 1
' Pozor na zacyklení Do bez podmínky
Do
    Cells(Pocitadlo + 4, 1) = Pocitadlo
    Pocitadlo = Pocitadlo + 1 ' Zvyš počítadlo.
    If Pocitadlo > 5 Then
        Exit Do
    End If
    '' MsgBox (pocitadlo)
Loop
```

1.7.2.7 Použití While Wend

Ukázka podobného použití oproti smyčce Do While Loop pro zápis čísla 1 až 10, do příslušných buněk (od řádku 5 ve sloupci A). V případě potřeby můžete k zobrazení čísel využít i dialogové okno **MsgBox**.

```
Pocitadlo = 1
While Pocitadlo <= 10 ' Vnitřní smyčka.
    Cells(Pocitadlo + 4, 1) = Pocitadlo
    Pocitadlo = Pocitadlo + 1 ' Zvyš počítadlo.
    '' MsgBox (pocitadlo)
Wend
```

1.7.2.8 Vypsat vybraný pro měsíc a rok do sloupce dny od prvního po poslední

Zde je krásně demonstrováno, že nemůžete vědět kolik dnů vybraný měsíc má, proto se kontroluje zda měsíc číselné řady spadá do vybraného měsíce.

```
Dim KonkretniDatum As Date
Dim i As Integer
Dim Mesic As Integer
Dim Rok As Integer
Rok = 2021
Mesic = 8
i = 0
KonkretniDatum = DateSerial(Rok, Mesic, 1)
Do While Month(KonkretniDatum) = Month(Date)
    Range("A1").Offset(i, 0) = KonkretniDatum
    i = i + 1
    KonkretniDatum = KonkretniDatum + 1
Loop
```

1.8 Konstrukce With

Jak efektně vytvořit posloupnost příkazů pro jeden objekt.

1.8.1 Syntaxe

```
With objekt
    [příkazy]
End With
```

1.8.2 Praktické ukázky

Jak prakticky využít konstrukci With

1.8.2.1 Nastavit více formátu nad jednou buňkou

Potřebujete v listě List1 formátovat buňku A1. Můžete využít konstrukci With. Další možností řešení může být **Setování**. Jak na **formát buněk** v samostatné kapitole.

```
With Worksheets("List1").Range("A1")
    .Value = 30
    .Font.Bold = True
    .Interior.Color = RGB(255, 255, 0)
End With
```

Aneb náhrada méně přehlednějšího a pomalejšího použití v kódu

```
Worksheets("List1").Range("A1").Value = 30
Worksheets("List1").Range("A1").Font.Bold = True
Worksheets("List1").Range("A1").Interior.Color = RGB(255, 255, 0)
```

1.8.2.2 Nastavení formátu pro aktivní buňku

Podobně můžete nastavit formát pro aktivní buňku

```
With ActiveCell.Font
    .Bold = True
    .Color = vbBlue
    .Name = "Arial"
    .Size = 22
    .Italic = True
End With
```

1.8.2.3 Vnoření konstrukce With

I konstrukci With můžete vnořovat:

```
With Workbooks("MujSešit").Worksheets("List1").Cells(1, 1)
    .Formula = "=SQRT(25)"
    With .Font
        .Name = "Arial"
        .Bold = True
        .Size = 8
    End With
End With
```

2 Obsah

1	ZÁKLADNÍ KONSTRUKCE PRO ŘÍZENÍ TOKU	1
1.1	Podmínková konstrukce If ... Elseif ... Else ... Then	1
1.1.1	Syntaxe	1
1.1.1.1	Operátory	2
1.1.2	Praktické ukázky If Then	2
1.1.2.1	Odpověď ano/ne na dotaz v dialogovém okně (MsgBox)	2
1.1.2.2	Větvení na základě více podmínek - věk	3
1.1.2.3	Provedení po splnění všech nebo jedné a více podmínek (OR – AND)	3
1.1.2.4	Více podmínek a možné komplikace - závorky	4
1.1.2.5	Dvě podmínky a ošetřit všechny možnosti	4

1.1.2.6	Vnořování podmínek	5
1.1.2.7	Ošetření chyb	6
1.2	Podmínková konstrukce – funkce IIF	6
1.2.1	Syntaxe Iif	6
1.2.2	Praktická ukázka použití Iif	6
1.3	Podmínková konstrukce Select Case	6
1.3.1	Syntaxe Select - Case	6
1.3.1.1	Operátory	6
1.3.2	Praktické ukázky použití Select Case	7
1.3.2.1	Kontrola dvou hodnot	7
1.3.2.2	Příklad na rozsahy - známkování	7
1.3.2.3	Příklad na rozsahy od do – (Is)	7
1.3.2.4	Výčet hodnot – využitím čárky	8
1.3.2.5	Praktická ukázka rozsah a obsah textu	8
1.3.2.6	Ukázka reakce na tlačítka v dialogovém okně	8
1.4	Cyklus For next	9
1.4.1	Syntaxe	9
1.4.1.1	Násilné Ukončení programu	9
1.4.2	Praktické použití cyklů For next	9
1.4.2.1	Základní cyklus od do	9
1.4.2.2	Zápis čísel do buněk	9
1.4.2.3	Násilné ukončení cyklu využitím - Exit For	9
1.4.2.4	Vnoření cyklu – dvojrozměrné pole	9
1.4.2.5	Kombinace cyklu a podmínek	10
1.4.2.6	Zápis do oblastí	10
1.4.2.7	Vyplnit náhodná čísla ve matici buněk (sloupce, řádku)	10
1.4.2.8	Podbarvit každý třetí řádek	10
1.4.2.9	Šachovnice	11
1.4.2.10	Úkol je potřeba mít od zadaného měsíce mínus 3 měsíce	11
1.4.2.11	Úkol je potřeba mít od zadaného měsíce plus 3 měsíce	11
1.4.2.12	Hody kostkou	12
1.4.2.13	Výpis hodnot v pole (Array)	12
1.5	Odskoky – větvení programu využitím Goto	12
1.5.1	Praktická ukázka použití GoTo	12
1.5.1.1	Odskoky na demonstraci možných problému	12
1.5.1.2	Práce s chybami	13
1.6	For ... Each	13
1.6.1	Syntaxe	13
1.6.2	Prakticky	13
1.6.2.1	Výpis všech listů v sešite	14
1.6.2.2	Zavřít všechny otevřené sešity	14
1.6.2.3	Smazat všechny objekty v aktivním listě	14
1.6.2.4	Výpis hodnot buněk v oblasti	14
1.6.2.5	Násilné ukončení cyklu For Each využitím Exit For	14
1.6.2.6	Vyplnění oblastí pro porovnání s řešením For Next	15
1.7	Do Loop Until	15
1.7.1	Syntaxe	15

1.7.1.1	Syntaxe s vybranou podmínkou na začátku	15
1.7.1.2	Syntaxe s vybranou podmínkou na konci	15
1.7.1.3	Pouze Do Loop s podmínkou	15
1.7.1.4	Ekvivalent WhileWend	16
1.7.2	Do Loop – prakticky	16
1.7.2.1	Počítadlo Do While s podmínkou na začátku	16
1.7.2.2	Počítadlo Do While s podmínkou na konci	16
1.7.2.3	Počítadlo Do Until s podmínkou na začátku	16
1.7.2.4	Počítadlo Do Until s podmínkou na konci	16
1.7.2.5	Zápis do buněk text a číslo	16
1.7.2.6	Do Loop s „násilným“ ukončením	17
1.7.2.7	Použití While Wend	17
1.7.2.8	Vypsát vybraný pro měsíc a rok do sloupce dny od prvního po poslední	17
1.8	Konstrukce With	17
1.8.1	Syntaxe	17
1.8.2	Praktické ukázky	18
1.8.2.1	Nastavit více formátu nad jednou buňkou	18
1.8.2.2	Nastavení formátu pro aktivní buňku	18
1.8.2.3	Vnoření konstrukce With	18
2	OBSAH	18