

Při výkladu funkcí (Function) ve VBA se na první místě zmiňuje, že na rozdíl od procedur (Sub) vracejí hodnotu a vyskytují se tak často na pravé straně příkazů přiřazení. A stejně jako procedury, které „tajně“ mohou propašovat ven hodnoty prostřednictvím veřejných proměnných nebo přebírání argumentů odkazem (ByRef), tak i funkce mají svá tajemství. Pokud je užijeme na listu, stávají se „dospělými“ uživatelskými funkcemi (UDF) – a měly by se krotit ve svém chování. Jejich výsledky neopouští místo, odkud jsou volány, tzn. buňky (viz Application.Caller), a jejich užití neovlivňuje ani prostředí aplikace. Je tomu tak vždy? Pojdme se podívat na jejich čertovské kousky, když se nikdo nedívá...

MsgBox, InputBox, FileDialog, ...

1	Function TESTOVKA()
2	'dialog (MsgBox)
3	TESTOVKA = MsgBox("Baf!")
4	End Function



Vlastní funkce - MsgBox

1	Function TESTOVKA() As String
2	'dialog s textovým polem (InputBox)
3	TESTOVKA = Application.InputBox("Kdo jsi?", "Identifikace")
4	End Function



Vlastní funkce - InputBox

1	Function TESTOVKA()
2	'dialog pro výběr souboru
3	With Application.FileDialog(msoFileDialogFilePicker)
4	.AllowMultiSelect = False
5	.Show
6	'vrací cestu k vybranému souboru
7	TESTOVKA = .SelectedItems(1)
8	End With
9	End Function



Vlastní funkce – FileDialog

CommandBars a Dialogs

1	Function TESTOVKA()
2	'zobrazení panelu
3	'nestandardní chování po uzavření
4	'zde Vložit symbol
5	Application.CommandBars.FindControl(ID:=308).Execute
6	TESTOVKA = "Hej!"
7	End Function

Vyvolávání panelů ve funkcích může způsobit neočekávané stavy. Excel často v danou chvíli jednoduše neočekává jejich zobrazení a neumí si poradit se sledem událostí v souvislosti s editací buňky. Vestavěné dialogy kolekce Dialogs se mi nepodařilo z funkcí vyvolat.

Komentáře

1	Function TESTOVKA()
2	'komentář v buňce, lze nastavit i pro jinou buňku
3	Application.Caller.AddComment "Jupí!"
4	TESTOVKA = "Jejda!"
5	End Function



Vlastní funkce – komentář

Zjednodušeně lze říci, že vykreslované objekty lze ve funkcích na listu použít. A ano, takový komentář lze například vložit do sousední buňky (Application.Caller.Offset(0, 1))

Ověření

```

1  Function TESTOVKA()
2  'ověření, lze nastavit i pro jinou buňku
3  With Application.Caller.Validation
4      .Delete
5      .Add Type:=xlValidateList, AlertStyle:=xlValidAlertInformation, _
6          Operator:=xlBetween, Formula1:="první;druhá"
7      .IgnoreBlank = True
8      .InCellDropdown = True
9      .InputTitle = ""
10     .ErrorTitle = ""
11     .InputMessage = ""
12     .ErrorMessage = ""
13     .ShowInput = False
14     .ShowError = False
15 End With
16     TESTOVKA = "Šmarjá!"
17 End Function

```

Pro ověření samozřejmě platí, že výsledek funkce nesmí být v rozporu s pravidly ověřování.



Vlastní funkce - ověření

Obrazce

```

1  Function TESTOVKA()
2  'nakreslení tvaru
3  ActiveSheet.Shapes.AddShape msoShapeCorner, 114, 30, 96, 90
4  TESTOVKA = "Nakresleno!"
5  End Function

```



Vlastní funkce - obrazec

Grafy

```

1 Function TESTOVKA()
2   'vložení grafu
3   ActiveSheet.Shapes.AddChart.Select
4   ActiveChart.ChartType = xlColumnClustered
5   ActiveChart.SetSourceData Source:=Range("List1!$C$7:$E$8")
6   TESTOVKA = "Safra!"
7 End Function

```



Vlastní funkce - graf

Barva pozadí a ohraničení buňky

Nenašel jsem způsob, jak ve funkci úspěšně nastavit barvu pozadí buňky a to ani podmíněným formátem, který by jinak aplikovat šel. Omezené možnosti má i ohraničení buňky.

```

1 Function TESTOVKA()
2   'ohraničení, nelze měnit typ a tloušťka čáry
3   With Application.Caller.Borders(xlDiagonalUp)
4     .LineStyle = xlContinuous
5     .ColorIndex = 3
6   End With
7   TESTOVKA = "Jaj!"
8 End Function

```



Vlastní funkce - ohraničení

Jiné buňky, výběry, odskoky, aktivace

Ve funkcích listu nelze měnit výběry a provádět aktivace objektů (.Select, .Activate), ani řídit odskoky přes Application.GoTo nebo hypertextové odkazy. Zakázány jsou také zápisy do oblastí buněk (Range, Cells) nebo manipulace s definovanými názvy (pojmenovanými vzorci).

Změny titulků, popisků aj.

Jednoznačně není možné říci, co lze a nelze měnit. Kupříkladu stavový řádek se ovlivnit z funkce listu

nenechá, na druhou stranu titulek aplikace ano.

1	Function TESTOVKA()
2	Application.Caption = "Můj šéf je lump!"
3	TESTOVKA = "Hmm!"
4	End Function

Filtry, Tabulky

Z funkcí listu není možné si pohrát s automatickým filtrem nebo třeba vložit Tabulku (Seznam). Na druhou stranu vás nechá setřídít oblast.

1	Function TESTOVKA()
2	'setřídění oblasti
3	With ActiveWorkbook.Worksheets("List1").Sort
4	.SetRange Range("A1:A6")
5	.Header = xlNo
6	.MatchCase = False
7	.Orientation = xlTopToBottom
8	.SortMethod = xlPinYin
9	.Apply
10	End With
11	TESTOVKA = "Aha!"
12	End Function

Volání procedury

Funkce se může odvolávat na jiné funkce a procedury. V tu chvíli pro ně platí stejná pravidla jako pro funkce, ze kterých jsou volány. Ve funkcích je stejně jako v procedurách možné užít v rámci parametrů ByRef, nicméně z „funkčního vězení“ ven nepropašujete žádný moták s hodnotou.

1	Function TESTOVKA()
2	'volání procedury
3	Call Procedura
4	TESTOVKA = "Sláva!"
5	End Function
6	
7	Sub Procedura()
8	MsgBox "Hurá!"
9	End Sub



Vlastní funkce – volání procedury

Nelze načasovat spuštění procedury (Application.OnTime), ani ve funkci přiřadit makro klávesové zkratce (Application.OnKey). Ve funkci listu není možné nastavit události (Application.OnEntry, .OnDoubleClick, .OnCalculate aj.). Excel 4 makra (ExecuteExcel4Macro) jsou rovněž odstaveny na druhou kolej.

Průšvihy v zabezpečení

Ve funkcích listu je možné uplatnit metodu SendKeys pro odesílání virtuálních stisků kláves.

1	Function TESTOVKA()
2	'text následovaný stiskem ENTER
3	SendKeys "Jde to!~"
4	TESTOVKA = "Vida!"
5	End Function



Vlastní funkce – SendKeys

Sekvence je odeslána až po vrácení hodnoty a ukončení editace buňky.

1	Function TESTOVKA()
2	Dim strVzorec As String
3	'vložení vzorec do buňky vpravo
4	strVzorec = Application.Caller.FormulaLocal
5	SendKeys "{UP}{TAB}" & strVzorec & "~"
6	TESTOVKA = "Vzorec vpravo!"
7	End Function



Vlastní funkce – SendKeys podruhé

Projde i sebedestrukce: SendKeys „{UP}^-~“.

Šokem byla pro mě snadná cesta spuštění prakticky libovolné aplikace přes funkci Shell.

```
1 Function TESTOVKA()  
2 'spuštění aplikace (Poznámkový blok) bez její aktivace  
3 ID = Shell("Notepad", vbMinimizedNoFocus)  
4 TESTOVKA = "ID"  
5 End Function
```

Excel neklade odpor ani při aktivaci nainstalovaného doplňku.

```
1 Function TESTOVKA()  
2 'spuštění doplňku  
3 AddIns("Analytické nástroje").Installed = True  
4 TESTOVKA = "Chi chi!"  
5 End Function
```



Vlastní funkce - aktivace doplňku

Ve funkcích listu je do jisté míry přípustné používat API.

```
1 Private Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" _  
2 (ByVal lpBuffer As String, nSize As Long) As Long  
3  
4 Function TESTOVKA()  
5 'výpis aktuálního uživatele Windows  
6 Dim strUzivatel As String  
7 strUzivatel = String(100, Chr$(0))  
8 GetUserName strUzivatel, 100  
9 strUzivatel = Left$(strUzivatel, InStr(strUzivatel, Chr$(0)) - 1)  
10 TESTOVKA = strUzivatel  
11 End Function
```

```
1 Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)  
2  
3 Function TESTOVKA()  
4 'pozastavení kódu na 5 sekund  
5 Sleep 5000  
6 TESTOVKA = "API!"  
7 End Function
```

```

1  Private Const SND_ASYNC = &H1
2  Private Const SND_NODEFAULT = &H2
3
4  Private Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" _
5      (ByVal lpszSoundName As String, ByVal uFlags As Long) As Long
6
7  Function TESTOVKA()
8      'přehrání zvuku (asynchronně, nečeká na ukončení procedury)
9      sndPlaySound "c:\Windows\Media\notify.wav", SND_ASYNC Or SND_NODEFAULT
10     TESTOVKA = "Tutú!"
11 End Function

```

Ukázali jsme si, že vlastní funkce nejsou až takové puritánky, za jaké je učebnice Excelu vydávají. Dnešní cvičení berte ovšem za čistě experimentální. V řadě případů vám nikdo nezaručí, jak se Excel zachová poté, co mu hodíte na záda vykutálenou funkci. Problémy mohou nastat při přepočtu (viz také Application.Volatile), při volání událostí ze zásobníku atd.