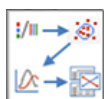




# Programujeme v softwaru Statistica

díl čtvrtý

Newsletter Statistica ACADEMY



Téma: Programování, makra, skripty  
Typ článku: Návody

*Tento článek navazuje na sérii předchozích článků, ze kterých už víme, jak si napsat nebo zaznamenat makro Statistica. Dnes se zaměříme na úpravy vygenerovaných maker, ukážeme si v nich místa, která jsou vhodná pro úpravy, a taky si popíšeme důležité objekty, které se v makrech Statistica často vyskytují a jejichž znalost je „povinná“.*

Jak už víme z minulých článků, software Statistica nabízí možnost generování maker v jazyku SVB (Statistica Visual Basic). Makro by mělo fungovat již po vygenerování, ale obvykle funguje pouze pro danou situaci, pro kterou bylo vygenerováno. Problém může nastat později, pokud takové makro spustíme např. na souboru dat s jinou strukturou, než pro který bylo vygenerováno. Pojďme se tedy podívat na strukturu vygenerovaného makra, na kterém si ukážeme důležitá místa, vhodná pro úpravy.

## Úprava generovaných maker

Makro vygenerujeme na ukázkovém souboru **Adstudy.sta**, z menu **Soubor -> Otevřít příklady -> Datasets -> Adstudy.sta**.

Vytvoříme analýzu základních statistik, menu **Statistiky -> Základní statistiky -> Popisné statistiky**, zvolíme proměnné **3-6 (MEASURE01-MEASURE 04)**, a zobrazíme výsledek tlačítkem **Výpočet**. Makro SVB (Statistica Visual Basic) vygenerujeme přes pravé tlačítko na dolním panelu, kde běží minimalizovaná analýza -> **Vytvořit makro**. Všechny způsoby záznamu jsou popsány v jednom z [předchozích článků](#). Vygenerované makro si pojmenujeme a necháme výchozí nastavení. Získáme následující kód:

```

'#Language "WWB-COM"
Option Base 1
Sub Main

    Dim newanalysis As Analysis
    Set newanalysis = Analysis (scBasicStatistics, ActiveInputDataSet)
    Dim oStaDocs As StaDocuments

    ' Základní statistiky a tabulky: Adstudy.sta
    Dim oAD1 As STABasicStatistics.BasStartup
    Set oAD1 = newanalysis.Dialog
    oAD1.Statistics = scBasDescriptives

    newanalysis.Run

    ' Popisné statistiky: Adstudy.sta
    Dim oAD2 As STABasicStatistics.BasDescriptiveStatistics
    Set oAD2 = newanalysis.Dialog
    oAD2.Variables = "3-6"
    oAD2.PairwiseDeletionOfMD = True
    oAD2.DisplayLongVariableNames = False
    oAD2.ExtendedPrecisionCalculations = False
    oAD2.PlotMedianQuartileRange = False
    oAD2.PlotMeanSEAndSD = False
    oAD2.PlotMeanSD196TimesSD = True
    oAD2.PlotMeanSE196TimesSE = False
    oAD2.UserDefinedPercentiles = False
    oAD2.ValidN = True
    oAD2.PercentOfValidObservation = False
    oAD2.Mean = True
    oAD2.Median = False
    oAD2.Mode = False
    oAD2.GeometricMean = False
    oAD2.HarmonicMean = False
    oAD2.ConfLimitsForMeans = False
    oAD2.ConfLimitsForSD = False
    oAD2.TrimmedMean = False
    oAD2.WinsorizedMean = False
    oAD2.GrubbsTest = False
    oAD2.Sum = False
    oAD2.StandardDeviation = True
    oAD2.CoefficientofVariation = False
    oAD2.Variance = False
    oAD2.StandardErrorOfMean = False
    oAD2.MinimumMaximum = True
    oAD2.LowerUpperQuartiles = False
    oAD2.Range = False
    oAD2.QuartileRange = False
    oAD2.Skewness = False
    oAD2.Kurtosis = False
    oAD2.StandardErrorOfSkewness = False
    oAD2.StandardErrorOfKurtosis = False
    oAD2.UseNumberOfIntervals = True
    oAD2.NumberOfIntervals = 10
    oAD2.NormalExpectedFrequencies = False
    oAD2.KSAndLillieforsTestForNormality = True
    oAD2.ShapiroWilkWTest = False
    oAD2.ConfidenceIntervalForMeansPlot = 95
    oAD2.CompressedStemAndLeaf = False

    Set oStaDocs = oAD2.Summary
    newanalysis.RouteOutput(oStaDocs).Visible = True
    Set oStaDocs = Nothing

End Sub

```

Zajímavý je hned druhý řádek procedury *Main*:

```
Set newanalysis = Analysis (scBasicStatistics, ActiveInputDataSet)
```

Zde, mimo definice typu analýzy (*scBasicStatistics* pro základní analýzy), definujeme zdroj dat, v tomto případě pomocí vlastnosti *ActiveInputDataSet*. Jedná se o vlastnost aplikace Statistica, která vrací aktuální tabulku (ta nejvíce nahoře v případě více současně otevřených tabulek). Pokud bychom si otevřeli jinou tabulku Statistica a makro opět spustili, bude vykonáno právě na této tabulce (předpokládá se, že naposledy otevřená tabulka bude nejvíce nahoře). Pokud budou v této tabulce k dispozici proměnné 3-6 odpovídajícího datového typu (v tomto případě číselná proměnná), makro by se mělo v pořádku vykonat i na těchto datech. Je však vhodnější si tabulku nadefinovat pevně, nejlépe použít tabulku uloženou v počítači. Upravené makro by potom vypadalo následovně:

```
Dim Tabulka As Spreadsheet
Set Tabulka =Spreadsheets.Open("C:\Program Files\Dell\Statistica
13\Examples\Datasets\Adstudy.sta")
Dim newanalysis As Analysis
Set newanalysis = Analysis (scBasicStatistics, Tabulka)
```

Toto makro se bude vykonávat pouze na tabulce *Adstudy.sta*, uložené v adresáři s příklady Statistica.

Jak už bylo naznačeno, další důležitou informací je volba proměnných. Obvykle se se jedná o vlastnost *Variables* nějaké analýzy. V našem případě je analýza deklarována jako *oAD2*:

```
oAD2.Variables = "3-6"
```

Proměnné zadáváme jako textový řetězec, jednotlivé proměnné jsou odděleny mezerou. Můžeme použít i rozsah Od-Do, nebo kombinaci, např. „3 4-6“. Lze také použít název proměnné, např. "MEASURE05 MEASURE09".

Dále v makru vidíme spoustu řádků, které představují vesměs nastavení výchozích hodnot analýzy. Pokud bychom je smazali, makro by stále fungovalo a pro tyto smazané parametry by byly opět použity výchozí hodnoty. Makro by tedy mohlo vypadat následovně:

```
'#Language "WWB-COM"
Option Base 1
Sub Main

Dim Tabulka As Spreadsheet
Set Tabulka =Spreadsheets.Open("C:\Program Files\Dell\Statistica 13\Examples\Datasets\Adstudy.sta")

Dim newanalysis As Analysis
Set newanalysis = Analysis (scBasicStatistics, Tabulka)
Dim oStaDocs As StaDocuments

' Základní statistiky a tabulky: Adstudy.sta
Dim oAD1 As STABasicStatistics.BasStartup
Set oAD1 = newanalysis.Dialog
oAD1.Statistics = scBasDescriptives

newanalysis.Run

' Popisné statistiky: Adstudy.sta
Dim oAD2 As STABasicStatistics.BasDescriptiveStatistics
Set oAD2 = newanalysis.Dialog
oAD2.Variables = "3-6"

Set oStaDocs = oAD2.Summary
newanalysis.RouteOutput(oStaDocs).Visible = True
Set oStaDocs = Nothing

End Sub
```

Při spuštění redukovaného makra bychom dostali stejný výsledek, jako v předchozím případě.

Když se podíváme na konec makra, objevíme další, neméně zajímavé řádky. Vlastnost analýzy *Summary* obvykle odpovídá v analýze tlačítku *Výpočet*. Tento řádek tedy spouští výpočet dané analýzy a výsledek ukládá do objektu *oStaDocs*, který je typu *StaDocuments*. *StaDocuments* představuje kolekci dokumentů Statistica, kterými mohou být tabulky a grafy Statistica. Přístup k jednotlivým objektům kolekce je přes vlastnost *Item()*, kde v závorce uvádíme pořadový index objektu v kolekci. V našem příkladu si můžeme přes vlastnost kolekce *Count* (tedy *oStaDocs.Count*) zjistit, že kolekce obsahuje jeden objekt. A víme, že výstupem analýzy je tabulka. Můžeme si tedy tabulku z kolekce uložit do samostatné tabulky:

```
Dim Vystup As Spreadsheet
Set Vystup=oStaDocs.Item(1)
Vystup.Visible=True
```

Pozn.: Pozor na to, jak máme nastaven *Option base*, pokud by byl *Option base 0*, potom by byl první index v kolekci roven 0, tedy `Set Vystup=oStaDocs.Item(0)`. Obvykle taky není nutné vlastnost *Item* uvádět a lze použít zkrácený zápis `Vystup=oStaDocs(0)`.

Vytáhnout si tabulku z kolekce je vhodné v případě, že bychom chtěli s výsledkovou tabulkou dále pracovat a použít ji např. v jiné analýze. Pokud bychom chtěli pouze zobrazit výsledek analýzy, potom není potřeba s objekty kolekce nic dělat. O vše se postará vlastnost, kterou nalezneme na dalším řádku našeho makra, a tou je *RouteOutput*. Jedná se o vlastnost analýzy (*Analysis*), jejíž pomocí zobrazíme celý obsah kolekce *StaDocuments*, reprezentující výstup analýzy. *RouteOutput* se řídí nastavením správce výstupů, a dle jeho nastavení budou výstupy směřovány do samostatných oken nebo společného pracovního sešitu.

Výsledné makro tedy bude vypadat následovně:

```
'#Language "WWB-COM"
Option Base 1
Sub Main

    Dim Tabulka As Spreadsheet
    Set Tabulka =Spreadsheets.Open("C:\Program Files\Dell\Statistica 13\Examples\Datasets\Adstudy.sta")

    Dim newanalysis As Analysis
    Set newanalysis = Analysis (scBasicStatistics, Tabulka)
    Dim oStaDocs As StaDocuments

    ' Základní statistiky a tabulky: Adstudy.sta
    Dim oAD1 As STABasicStatistics.BasStartup
    Set oAD1 = newanalysis.Dialog
    oAD1.Statistics = scBasDescriptives

    newanalysis.Run

    ' Popisné statistiky: Adstudy.sta
    Dim oAD2 As STABasicStatistics.BasDescriptiveStatistics
    Set oAD2 = newanalysis.Dialog
    oAD2.Variables = "3-6"

    Set oStaDocs = oAD2.Summary
    newanalysis.RouteOutput(oStaDocs).Visible = True

    Dim Vystup As Spreadsheet
    Set Vystup=oStaDocs.Item(1)
    Vystup.Visible=True

    Set oStaDocs = Nothing

End Sub
```

Po spuštění makra získáme výstup ve formě pracovního sešitu (pokud máme nastaveno umístování výsledků ve správci výstupů na pracovní sešit) a jedné samostatné tabulky Statistica.

Výše popsaný postup můžeme zobecnit na většinu analýz. Důležité je upravit zdroj dat, rozsah proměnných a eventuálně si z kolekce objektů výstupu umět konkrétní objekt uložit do samostatné proměnné (objektu).

Takto nahranou a upravenou analýzu můžeme použít např. v cyklu pro opakovaný výpočet s obměnou určitých parametrů. Ukážeme si to na příkladu, kdy budeme Základní statistiky počítat postupně na proměnné 3-6. Použijeme k tomu naše finální makro z předchozího příkladu. *Sub Main* přejmenujeme na *Sub Basics*, do které budeme přenášet jeden parametr – pořadové číslo proměnné:

```
Sub Basics (Citac As Long)
```

Nad touto procedurou vytvoříme novou proceduru *Main* (povinná pro každé makro SVB), ze které budeme postupně volat proceduru *Sub Basics*. Ve *For* cyklu, který bude probíhat od 3 do 6, tedy *For i=3 to 6*, budeme volat proceduru *Basics*, které předáme číslo proměnné, pro kterou budeme chtít vypočítat proceduru *Basics*. Všimněme si, jak jsme definovali v proceduře *Basics*, tedy kódu pro výpočet základních statistik, proměnné, pro které chceme výpočet provést:

```
oAD2.Variables = CStr(Citac)
```

Jelikož jsou proměnné v makru definovány textovým řetězcem, musíme použít konverzní funkci *CStr* pro převod čísla na text. Výsledné makro bude vypadat následovně:

```
'#Language "WwB-COM"
Option Base 1

Sub Main
    For i= 3 To 6
        Basics(i)
    Next
End Sub

Sub Basics (Citac As Long)
    Dim Tabulka As Spreadsheet
    Set Tabulka =Spreadsheets.Open("C:\Program Files\Dell\Statistica 13\Examples\Datasets\Adstudy.sta")

    Dim newanalysis As Analysis
    Set newanalysis = Analysis (scBasicStatistics, Tabulka)
    Dim oStaDocs As StaDocuments

    ' Základní statistiky a tabulky: Adstudy.sta
    Dim oAD1 As STABasicStatistics.BasStartup
    Set oAD1 = newanalysis.Dialog
    oAD1.Statistics = scBasDescriptives

    newanalysis.Run

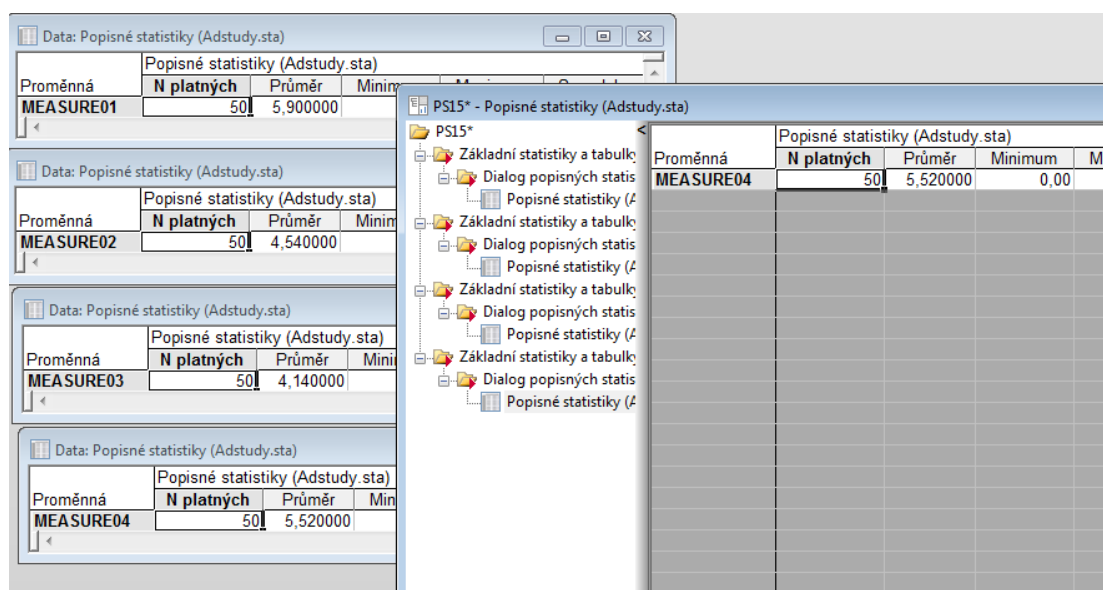
    ' Popisné statistiky: Adstudy.sta
    Dim oAD2 As STABasicStatistics.BasDescriptiveStatistics
    Set oAD2 = newanalysis.Dialog
    oAD2.Variables = CStr(Citac)

    Set oStaDocs = oAD2.Summary
    newanalysis.RouteOutput(oStaDocs).Visible = True

    Dim Vystup As Spreadsheet
    Set Vystup=oStaDocs.Item(1)
    Vystup.Visible=True

    Set oStaDocs = Nothing
End Sub
```

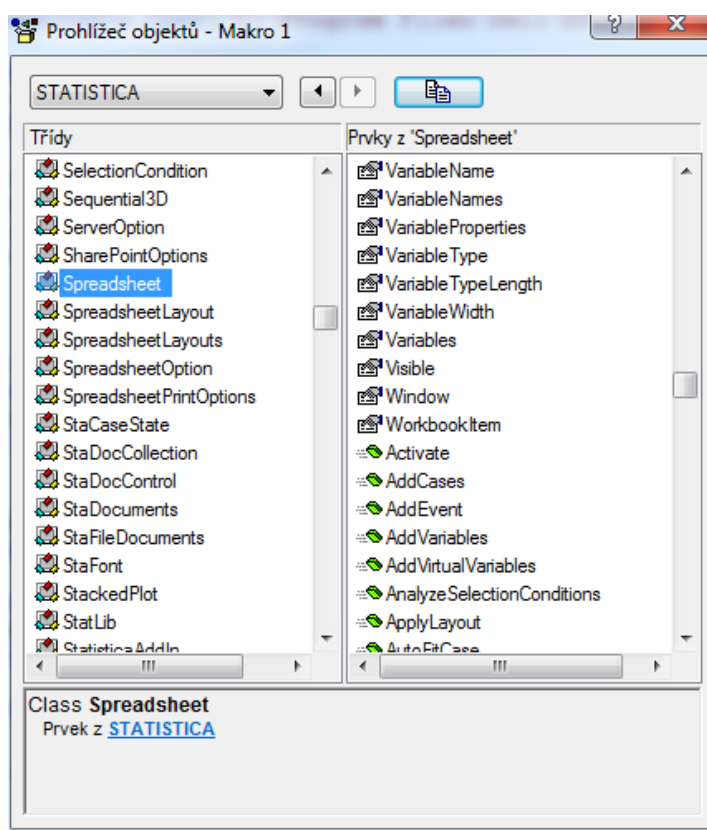
Výstupem budou 4 samostatné tabulky a jeden pracovní sešit (dle nastavení správce výstupů):



Výše popsaný příklad asi nebude mít smysl používat v praxi, jelikož lze jednoduše spočítat základní statistiky pro více proměnných najednou. Měl zejména demonstrovat, jak lze dynamicky měnit parametry opakovaných analýz.

## Tabulky Statistica - Spreadsheets

V tomto článku si ještě popíšeme jeden z nejzákladnějších objektů Statistica, se kterým se budeme při psaní maker setkávat – tím je tabulka Statistica, neboli *Spreadsheet*. Jedná se o objektovou proměnnou, jejíž třída je součástí základní knihovny *Statist.exe*. Obsahuje velkou řadu vlastností a funkcí, jejichž přehled je dostupný v prohlížeči objektů, třídě *Spreadsheet*.



Proměnná, která uchovává objekt tabulky, se deklaruje příkazem:

```
Dim Navez_tabulky As Spreadsheet
```

S takto nadeklarovanou tabulkou ale pracovat nemůžeme, jelikož se jedná o prázdný objekt, který je třeba inicializovat. Inicializaci můžeme provést již při deklaraci proměnné, a to za použití kouzelného slůvka *New*.

```
Dim Navez_tabulky As New Spreadsheet
```

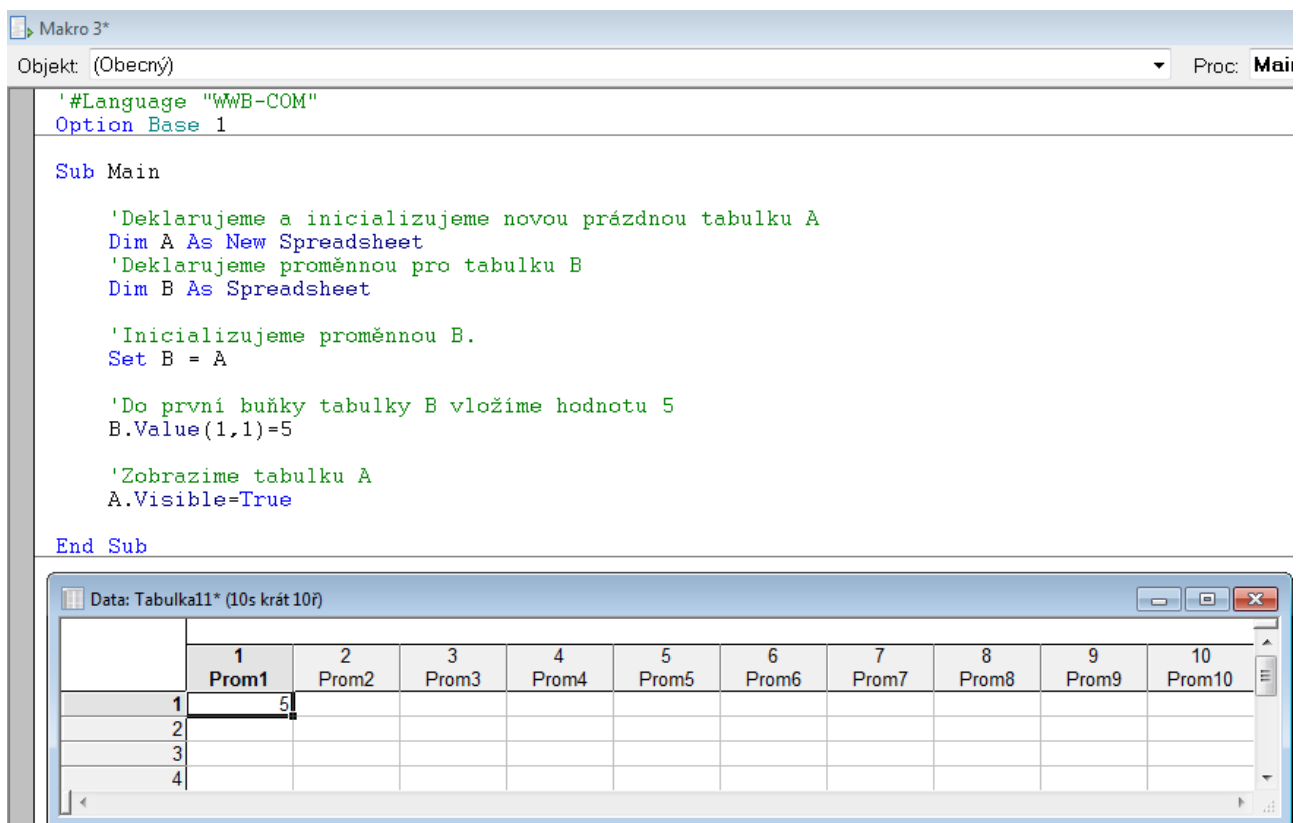
Takto získáme prázdnou tabulku s výchozími vlastnosti – velikost 10x10, proměnné typu double, atd.

Tabulku lze inicializovat i přiřazením již existující tabulky, např. výstupu nějaké analýzy, nebo otevřít tabulku z disku:

```
Dim Tab As Spreadsheet
```

```
Set Tab = Spreadsheets.Open("cesta_k_souboru_.sta")
```

Spreadsheet je objektová proměnná a u té víme, že se hodnoty předávají odkazem, tzn. pokud pracujeme s nějakou tabulkou a tu přiřadíme jiné proměnné, nevytváříme kopii a pouze nové proměnné ukážeme místo v paměti počítače, kde se tabulka nachází. Jedná se ale stále o tu samou tabulku. Pokud tabulku v rámci nové proměnné upravíme, tyto změny se projeví i u tabulky původní proměnné, ze které jsme přiřazovali. Nejlépe to ukáže následující příklad:



```
'#Language "VBA-COM"
Option Base 1

Sub Main

'Deklarujeme a inicializujeme novou prázdnou tabulku A
Dim A As New Spreadsheet
'Deklarujeme proměnnou pro tabulku B
Dim B As Spreadsheet

'Inicializujeme proměnnou B.
Set B = A

'Do první buňky tabulky B vložíme hodnotu 5
B.Value(1,1)=5

'Zobrazíme tabulku A
A.Visible=True

End Sub
```

|   | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10     |
|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
|   | Prom1 | Prom2 | Prom3 | Prom4 | Prom5 | Prom6 | Prom7 | Prom8 | Prom9 | Prom10 |
| 1 | 5     |       |       |       |       |       |       |       |       |        |
| 2 |       |       |       |       |       |       |       |       |       |        |
| 3 |       |       |       |       |       |       |       |       |       |        |
| 4 |       |       |       |       |       |       |       |       |       |        |

Vidíme, že po spuštění došlo ke změně v tabulce A, i když jsme ji inicializovali jako novou prázdnou tabulku. To proto, že v případě proměnné A i B se jedná tentýž objekt, na který obě proměnné odkazují (referují). Na to je třeba dát si pozor.

Zde výčet nejdůležitějších vlastností a funkcí pro *Spreadsheet*.

### Vlastnosti:

**NumberOfCases** – vrací počet případů tabulky

**NumberOfVariables** - vrací počet proměnných tabulky

**Value**(CaseNo As Long, VarNo As Long) – vrací/nastaví číselnou hodnotu buňky definované číslem případu a proměnné

**Text**(CaseNo As Long, VarNo As Long) As String – vrací textovou hodnotu buňky definované číslem případu a proměnné

**Name** – vrací/nastaví název tabulky

**Visible** – nastaví tabulku viditelnou. Výchozí hodnota je *False*, takže je doporučeno v případě potřeby změnit na *True*

**VariableName**(VarNo As Long) – vrací/nastaví název proměnné, definované jejím číslem

**Cells**(Row As Long, Column As Long) – nastaví kurzor na požadovanou buňku, zadanou číslem řádku a sloupce

**CellsRange**(FirstRow As Long, FirstColumn As Long, LastRow As Long, LastColumn As Long) – vybere obdélníkovou oblast buněk, zadanou počáteční (řádek x sloupec) a koncovou buňkou (řádek x sloupec)

**Saved** – nastaví tabulku jako uloženou, tedy jako by u ní neproběhly změny (u jejího názvu není hvězdička). Takou tabulku lze zavřít, aniž by se software dotázal, zda chceme změny uložit, což je při práci s makry výhodné

### Funkce:

**VariableNumber**(Name As String) – vrací číslo proměnné, definované jejím jménem

**SetSize**(CaseAmount As Long, VariableAmount As Long) – nastavení velikosti tabulky na počet případů x počet proměnných

**SaveAs**(Name As String, [Overwrite As Variant = 0]) – uloží tabulku do zadaného umístění; volitelný parametr přepsat v případě existující tabulky

**AddCases**(After As Long, HowMany As Long) – přidá prázdné případy, definované počtem (HowMany) případů, které chceme přidat, za který případ (After)

**AddVariables**(Name As String, After As Long, [HowMany As Variant], [Type As Variant], [Len As Variant], [MissingData As Variant], [Format As Variant], [LongName As Variant]) – přidá prázdné proměnné, definované jménem, počtem případů za který případ a dalšími volitelnými parametry, jako typ, řetězec pro prázdná data a další

## Závěrem

Generování maker je velmi užitečná funkce a ve spojení s drobnými úpravami můžeme dosáhnout efektu, který je někdy přes klasické nabídky nedosažitelný. Nabízí alternativu, která nemusí být pro všechny, ale své uplatnění si jistě najde, zejména v oblastech komplexnějších analýz, kde si můžeme řadu činností ulehčit a zautomatizovat.