

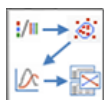
Programujeme v softwaru Statistica



Software

díl první

Newsletter Statistica ACADEMY



Téma: Programování, makra, skripty
Typ článku: Návody

Jistě jste se při práci v softwaru Statistica dostali do situace, kdy bylo zapotřebí vykonat hromadnou akci, jako např. spustit analýzu s variací různých parametrů, provést hromadnou úpravu dat v tabulce nebo aplikovat určitý styl na více grafů. Toho všeho lze dosáhnout za použití skriptů, které lze vytvářet v jazyce Statistica Visual basic.

Po přečtení prvních řádků řadu z vás možná napadlo, že tato možnost není nic pro mě. Pokud však máte alespoň základní představu o principech programování, Statistica nám v tomto ohledu jde naproti. Dokáže totiž hodně „špinavé“ práce odvést za nás a na nás bude výsledný produkt (v lepším případě) pouze lehce dodělat.

Vzhledem k tomu, že celá problematika může být velmi obsáhlá, nebudeme popisovat všechny podrobnosti programování, ale ukážeme si to základní, co je třeba vědět k tomu, abychom byli schopni vytvořit jednoduché makro. V některém z příštích článků si pak ukážeme, jak upravit a posléze použít makro vygenerované softwarem Statistica.

Jazyk, prostředí

Jak již bylo řečeno, nosným programovacím jazykem pro tvorbu skriptů je Statistica Visual basic. Jedná se o spojení standardního programovacího jazyka Visual basic, rozšířeného o knihovny Statistica. Tyto knihovny obsahují všechny funkcionality, dostupné prostřednictvím klasických nabídek. Teoreticky je tedy možné naskriptovat vše, na co jsme zvyklí při běžné práci v prostředí Statistica. Statistica Visual basic je kompatibilní s VBA (Visual Basic for Applications), tedy programovacím jazykem, který je používán v balíku Microsoft Office.

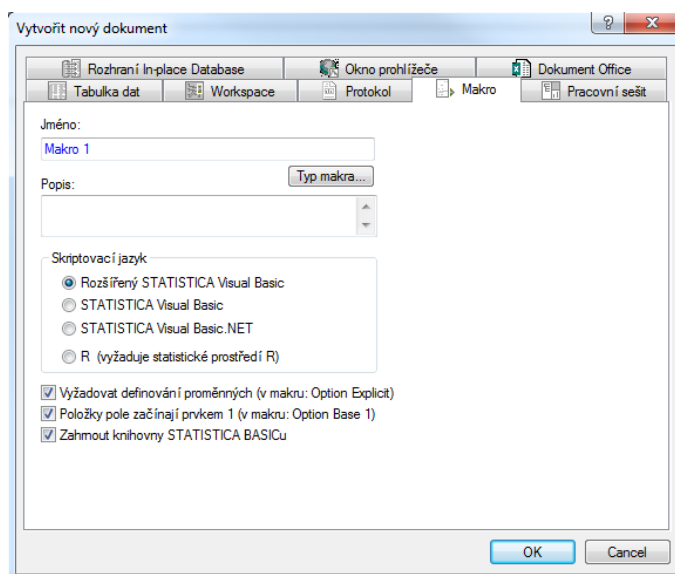
Skripty zapisujeme v rámci vyhrazeného souborového formátu, tzv. makra Statistica. Nový soubor makra vytvoříme v menu **Domů** -> **Nový** -> záložka **Makro**

Zde si vysvětlíme jednotlivé položky:

- **Jméno** – název makra, který se objeví v hlavičce okna a při ukládání souboru
- **Popis** - dodatečné uživatelské informace k makru; dostupné ve vlastnostech makra

Typ makra:

- **Standardní** – výchozí volba, představuje samostatně spustitelný kód makra
- **Kód** – vedlejší kód, který je uložen v samostatném souboru. Na tento soubor je třeba se odkázat ze standardního makra, (za použití klauzule #uses) a potom lze jeho jednotlivé části volat (používat) v rámci hlavního kódu. Jeho struktura se sestává z menších celků ve formě funkcí a procedur (viz níže)
- **Objekt** - vedlejší kód, který je uložen v samostatném souboru. Jeho struktura se sestává z objektů
- **Třída** - vedlejší kód, který je uložen v samostatném souboru. Jeho struktura se sestává z tříd



Skriptovací jazyk:

- **Rozšířený STATISTICA Visual basic** – jedná se o rozšíření Statistica Visual basic o některé funkce, datové typy a operátory, které nejsou kompatibilní s VBA. Pokud nebudeme makro přenášet mimo prostředí Statistica, můžeme zvolit tuto možnost
- **STATISTICA Visual basic** - spojení standardního programovacího jazyka Visual basic, rozšířeného o knihovny Statistica. Je kompatibilní s VBA.
- **STATISTICA Visual basic.NET** – programovací jazyk kompatibilní s Visual Basic.NET. Dovoluje používat externí knihovny, vytvořené v .NET jazyku, tzv. assembly. Tato možnost je dostupná, pokud je na počítači nainstalován .NET Framework 2.0.

Další možnosti dialogu:

- **Vyžadovat definování proměnných** – pokud je vybráno, je nutné každou proměnnou, před jejím použitím nadefinovat (nadeklarovat). V opačném případě je proměnná, při jejím prvním použití, automaticky nadefinována
- **Položky pole začínají prvkem 1** – říká, kterým indexem budou začínat proměnné typu pole. Pokud je vybráno, pole začíná indexem 1. V opačném případě začíná nulou.
- **Zahrnout knihovny STATISTICA BASICu** - pokud je vybráno, makro bude obsahovat odkaz na základní knihovny pro práci s grafy a základními statistickými a maticovými funkcemi.

Pro většinu maker nám postačí, když použijeme výchozí nastavení. V tomto příkladu s však pro názornost zaškrtneme všechny tři poslední možnosti dialogu. To proto, abychom si ukázali, jak se to promítne do výsledného makra. Jednotlivé volby jsou zaneseny na začátek makra, ještě před samotnou výkonovou část kódu. Pokud některá z možností nebude uvedena, bude použita její výchozí hodnota:

```
'#Language "WB-COM"
Option Base 1
Option Explicit
#Uses "*STB.SVX"
#Uses "*GRAPHICS.SVX"

Sub Main
End Sub
```

Zatím to vypadá možná složitě, ale to jen z důvodů popisu všech možností, které makra nabízejí. Pro běžnou práci s makry si vystačíme s výchozím nastavením, kód budeme zapisovat a spouštět jako standardní typ makra. Při výchozím nastavení získáme následující základní strukturu:

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

Sub Main
End Sub
```

Procedury, funkce

Nyní se dostáváme k té zajímavější části, samotnému psaní kódu, skriptu, nebo chcete-li, programování.

Každý funkční program se skládá, zejména z důvodu přehlednosti, z několika dílčích celků, minimálně však alespoň z jednoho. Tyto dílčí celky označujeme jako procedury (podprogramy, anglicky subroutines) a každá z nich může řešit jiný problém, např. výpočet matematické operace, tvorba grafu, odeslání e-mailu apod. Procedura nevrací žádnou hodnotu, může pouze upravovat hodnoty jiných proměnných. Existují však i procedury, které hodnotu uchovávají, samy vystupují v podstatě jako proměnná, a označujeme je jako funkce. Procedury i funkce zapisujeme do uzavřených bloků, které jsou uvozeny a ukončeny klíčovými slovy. Klíčová slova jsou v textu makra automaticky zvýrazněna modrou barvou a nelze je použít v jiném kontextu než v tom, pro který jsou určena. Procedury/funkce mají následující strukturu:

Procedura:

```
Sub Volitelný_jednoslovný_název
```

```
End sub
```

Funkce:

```
Function Volitelný_jednoslovný_název as datový_typ
```

```
End function
```

Všimněme si, že u funkce navíc definujeme datový typ (nemusíme, ale je to doporučeno, stejně jako explicitně definovat typ proměnné), a to z toho důvodu, jak již bylo řečeno, funkce je nositelem hodnoty.

Řekli jsme si, že každé makro musí být složeno minimálně z jedné procedury. Ta je povinná a nese i vyhrazený název *Main*. Při spuštění makra se procedura *Main* začíná zpracovávat jako první, a pokud z ní neodkazujeme na jiné procedury, potom se vykoná jako jediná. A to i přesto, že kód obsahuje procedur více. Odkazování na jiné procedury/funkce říkáme „volání“. Volání si prakticky předvedeme později.

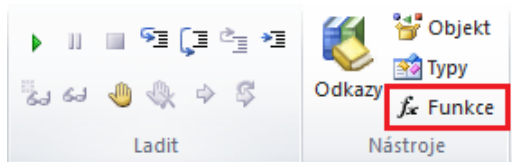
Proměnné

Proměnné nám v programu slouží k uchování hodnoty. Je dobrým zvykem proměnnou, než do ní uložíme hodnotu, nadefinovat (vytvořit), tzv. nadeklarovat. Deklarací říkáme, jaký typ informací proměnná ponese, tzn. např. text, číselné hodnoty, binární data a další. Pokud proměnnou před jejím prvním použitím nadeklaruje, potom se typ určí automaticky, dle ukládané hodnoty. Nicméně potom ztrácíme přehled o tom, co je kde uloženo a je vhodné se tomuto postupu vyhnout. S tím také souvisí jedna z možností při vytváření souboru makra – a to možnost „Vyžadovat definování proměnných“. Pokud je tato možnost vybrána, potom je třeba proměnnou před jejím prvním použitím vždy nadeklarovat. Pokud tak neučiníme, program, při jeho spuštění, nahlásí chybu. Proměnné si můžeme rozdělit na:

- **Základní** – nestrukturované, uchovávají jeden druh informace
 - *Integer* – celočíselná
 - *Double* – desetinné číslo
 - *String* – řetězec textu
 - *Boolean* - binární informace; může nabývat pouze hodnot (0,1) nebo též (False,True)
- **Složené**
 - pole, které uchovává sekvenci hodnot stejného druhu, na které se odkazujeme pořadovým indexem
- **Objektové** – složená struktura, která může nést více druhů informací
 - *Object* – obecná objektová proměnná

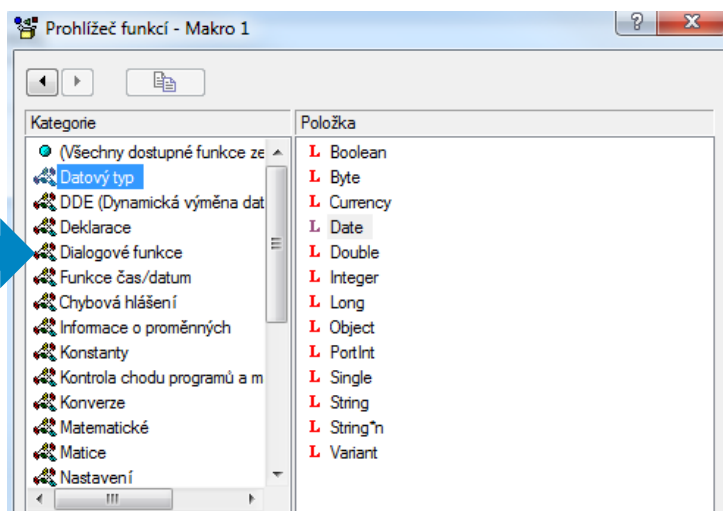
Přehled všech datových typů naleznete v prohlížeči funkcí – při otevřeném okně **makra**, záložka **Makro - > Funkce**. Prohlížeč funkcí je vůbec užitečná pomůcka, kde nalezneme popis spousty dalších prvků, které lze při programování použít:

Deklaraci proměnné, tedy její vytvoření, provádíme pomocí vyhrazeného slova **Dim** a **as**:



Dim *Nazev_promenne* **as** *typ_promenne*

Objektové proměnné je ještě potřeba, před jejich použitím, inicializovat. Inicializace proměnné si můžeme představit tak, že



proměnnou naplníme hodnotami, byť nulovými nebo výchozími. Výchozí hodnoty obvykle objekt nabyde při inicializaci objektu za použití klíčového slova **New**. Objektová proměnná, která není inicializovaná, má hodnotu „Nothing“ a pokud budeme chtít s takovou proměnnou pracovat, zpracování kódu se zastaví a zahlásí chybu.

Programujeme

Než se pustíme do psaní prvního makra, řekněme si několik pravidel pro zápis kódu:

- každý řádek reprezentuje jeden příkaz
- prázdný řádek je ignorován
- mezera může mít jakoukoli velikost
- textové řetězce zapisujeme mezi uvozovky „ ”
- komentář je uvozen apostrofem „ ’ ” ; text komentáře je automaticky obarven zeleně; komentář není zapojen do zpracování kódu

Pojďme si tedy vytvořit jednoduché makro, které nám sečte hodnoty dvou proměnných a výsledek vypíše na obrazovku. Vytvoříme si nové makro, záložka **Domů** -> **Nový** -> **záložka Makro**. Dialog s výchozími hodnotami potvrdíme.

Nadeklarujeme si dvě celočíselné proměnné, A a B:

```
Dim A as Integer
```

```
Dim B as Integer
```

Základní datové typy není třeba inicializovat a měly by již po deklaraci obsahovat nějakou výchozí hodnotu, obvykle 0 pro číselné, prázdný řetězec pro textové proměnné. Nespoléhejme však na to, jaké výchozí hodnoty obsahují a uložme si do proměnných výchozí hodnoty vlastní, ihned po jejich deklaraci. Přiřazení hodnoty provedeme symbolem „rovná se“ (=):

```
A=1
```

```
B=5
```

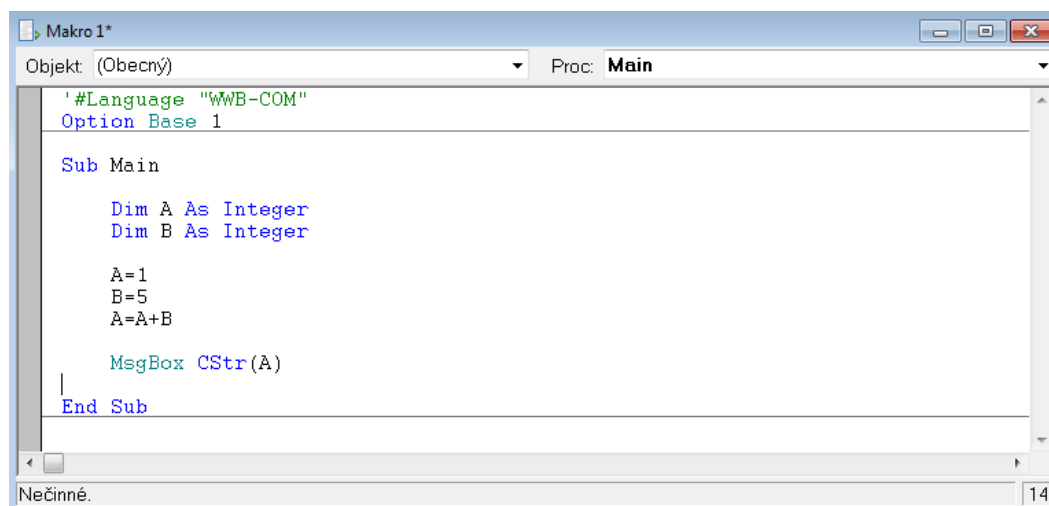
Nyní hodnoty sečteme. Výsledek však musíme uložit do nějaké proměnné. Buď si vytvoříme proměnnou novou, nebo hodnotu uložíme do proměnné již existující. K tomu můžeme dokonce použít i ty proměnné, které se účastní dané operace. V našem případě uložíme výsledek do proměnné A:

```
A=A+B
```

Pro kontrolu výsledku si necháme obsah proměnné A vypsát na obrazovku. Použije k tomu funkci **Msgbox**. Vzhledem k tomu, že tato funkce pracuje pouze s textovými proměnnými (textovými hodnotami), bude potřeba si ještě naši celočíselnou hodnotu převést na text, a to za pomoci funkce **Cstr()**:

```
Msgbox Cstr(A)
```

Celé makro potom vypadá následovně:



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

Sub Main

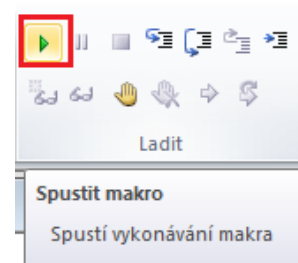
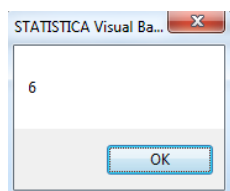
    Dim A As Integer
    Dim B As Integer

    A=1
    B=5
    A=A+B

    MsgBox CStr(A)

End Sub
```

Makro spustíme klávesou F5 nebo **zelenou šipkou** v menu, záložka **Makro**. Výsledek bude následující:



Abychom se konečně dostali i k praktickému využití objektů pro Statisticu, pojďme si vytvořit jednoduché makro, které vytvoří novou tabulku Statistica.

Vytvoříme nové makro, **Domů** -> **Nový** -> **záložka Makro**

Jako první si nadeklaruje proměnnou, ve které bude uložena tabulka Statistica. Bude se jednat o objektovou proměnnou typu **Spreadsheet**:

Dim Tabulka **as Spreadsheet**

Pokud budeme chtít s tabulkou pracovat, víme, že jakožto objekt ji musíme nejdříve inicializovat. To lze provést dvěma způsoby:

1. Řadu objektových proměnných lze inicializovat již při jejich deklaraci, za použití klíčového slova **New**:

Dim Tabulka **as New Spreadsheet**

2. Přiřazení nějaké hodnoty dané objektové proměnné


V tomto případě bychom museli proměnné *Tabulka* přiřadit již existující objekt typu **Spreadsheet**. Nebo použít funkci, která produkuje objekt typu **Spreadsheet**. Ukážeme si druhý příklad, tedy funkci, která vytvoří novou tabulku. Tato funkce je součástí kolekce *Spreadsheets*.

Kolekci si můžeme představit jako speciální pole objektů stejného typu. V tomto případě se jedná o kolekci všech tabulek v aktuální instanci Statisticu. Kolekce může mít doplňkové parametry, které lze číst, ukládat v nich informace. Těmto parametrům říkáme *Vlastnosti* (properties), a nebo *Funkce* (functions), které vytváří určitou akci. *Vlastnosti* nebo *Funkce* se zapisují za název kolekce, oddělené

tečkou. Můžeme si všimnout, že v okamžiku zápisu tečky, prostředí editoru maker nabídne všechny dostupné metody a funkce pro danou kolekci:

Pro vytvoření nové tabulky použijeme funkci *New*. Tady, na rozdíl od jednoduchých datových typů, kde stačilo pro přiřazení použít rovnítko, musíme před názvem **objektové proměnné** použít klíčové slovo **Set**:

```
Sub Main
    Spreadsheets.
End Sub
```



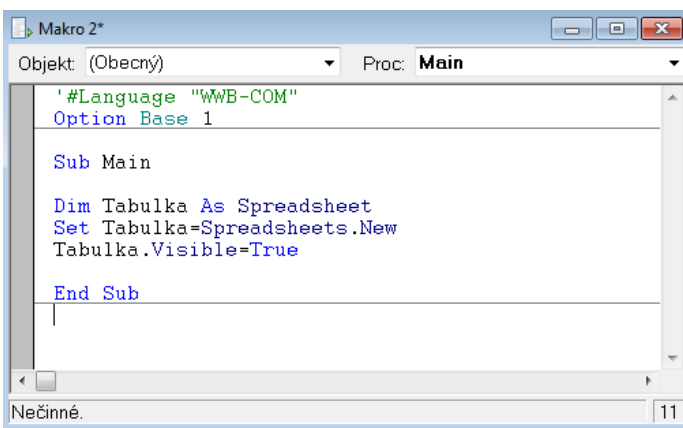
`Dim Tabulka as Spreadsheet`

`Set Tabulka=Spreadsheets.New`

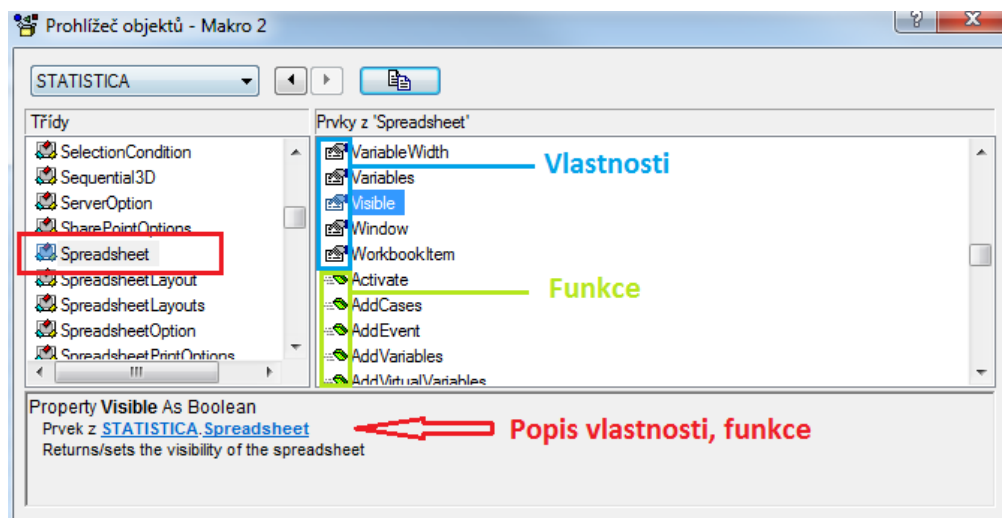
Pokud bychom tento kód spustili, makro by se zpracovalo, ale novou tabulku bychom neviděli. Proč? Tabulka se opravdu fyzicky vytvoří, jen se nám nezobrazí a zůstane skryta na pozadí. Objekty (objektové proměnné) mají, podobně jako kolekce, také své vlastnosti a funkce. Kolekce je totiž zase jen objekt. Jednou z těchto vlastností, pro tabulku Statistica, je vlastnost *Visible*. Jako výchozí hodnota je nastavena na 0, neboli *False*. Abychom tabulku zobrazili (zviditelnili), nastavíme vlastnost na hodnotu 1, nebo taky *True*:

`Tabulka.visible=True`

Výsledné makro vypadá následovně:



Když už byla řeč o vlastnostech a funkcích, doporučuji si je projít pro objekt *Spreadsheet*. Jsou dostupné v prohlížeči objektů, při otevřeném okně makra, záložka **Makro -> Objekt**.



V prohlížeči objektů si můžeme všimnout, že jednotlivé objekty v levém sloupci nesou označení jako *Třídy*. Třída nám obecně popisuje nějaký objekt, kterým může být již zmíněná tabulka Statistica, kolekce objektů, atd. Inicializací třídy získáme „fyzický“ objekt, se kterým můžeme pracovat. Některé třídy jsou inicializovány již při spuštění softwaru Statistica, jako např.

- *Application* – obsahuje objekt aplikace Statistica
- *Spreadsheets* – kolekce aktuálních tabulek Statistica
- *Workbooks* – kolekce aktuálních sešitů Statistica
- *Macros* – kolekce aktuálních maker Statistica

Závěrem

Tento článek popsal základní prvky, které je třeba si osvojit, pokud bychom chtěli začít software Statistica ovládat programově, pomocí maker Statistica. Do článku se vešel jen opravdový základ. V některém z příštích dílů bychom si ukázali, jak získat makra generovaná softwarem Statistica, používání procedur a funkcí a dalších důležitých objektů, které nám umožňují zase o něco sofistikovanější práci se softwarem Statistica.