Studijní opora

Seminář 2

Ing. Michal Dostál, Ph.D.
katedra informatiky EF TUL

Obsah

Financováno Evropskou unií NextGenerationEU
Národní plán obnovy
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

2 Úvod do jazyka VBA

2.1 Procedury

Procedura je ve VBA pojmenovaná část kódu, která obsahuje sekvenci tvrzení a výrazů, které budou provedeny. Příkladem může být funkce (typ procedury) nazvaná `Left` (vrátí levou část textu, kterou uživatel specifikuje). Řetězec `ahoj` má 4 znaky a pokud na něj aplikujeme danou funkci v tomto tvaru `Left("ahoj",3)`, vrátí nám výraz `aho`.

Veškerý spustitelný kód ve VBA musí být zapsán v procedůře.

2.2 Práce s editorem VBA

Jak jsme si na minulém cvičení ukázali, záznamník maker funguje tak, že veškeré akce, které při záznamu provedeme jsou zaznamenány pomocí syntaxe Visual Basic for Applications. V tomto jazyce můžeme vytvářet též vlastní kód, a to tří typů:

- procedury, které poznáme pomocí zápisu `Sub` (jsou používané stejným způsobem jako makra pořízená záznamem)
- uživatelské funkce (definované jako `Function`), které vkládáme do buněk jako vzorce a
- vlastnosti (`Property`) používané při tvorbě vlastních tříd.

Při zobrazení Editoru VB (záložka Vývojář a tlačítko **Visual Basic**) máme k dispozci okno, obsahující tři základní části.

První z nich je **podokno projektu** (na obrázku označeno jako 1), které obsahuje seznam listů v sešitu, položku **ThisWorkbook**, složku **Modules** nebo složku **Forms**, která obsahuje uživatelské formuláře.

Druhou část okna tvoří **podokno vlastností** (oblast 2). U modulů toto podokno zobrazuje pouze `Name`, tedy jeho název.

V třetí části okna najedeme hlavní okno pro zápis kódu. Na obrázku níže není otevřený žádný modul, proto je tato obalast zašedlá.

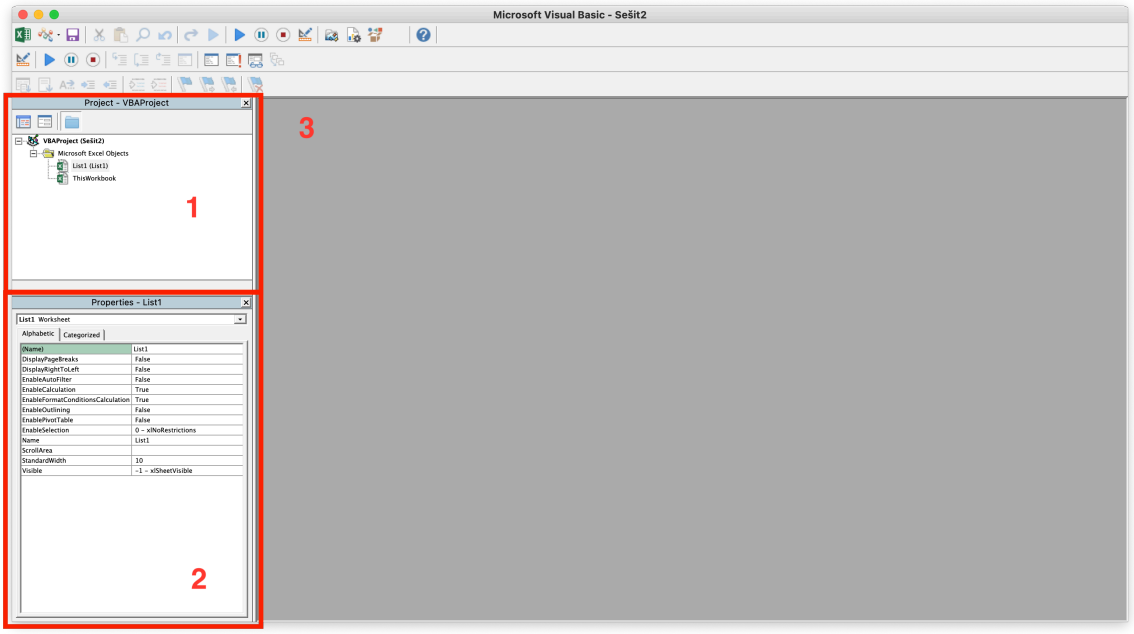
2.2.1 Vložení nového modulu

Abychom mohli začít psát nové makro, přidáme si nový modul pomocí volby **Instert > Module**. V podokně projektu se nám zobrazí nová položka `Module 1`. Když dvakrát poklepete na nově vytvořený modul, zobrazí se Vám v hlavním okně jeho obsah. V tuto chvíli je prázdný.

2.3 Základní syntaxe

Když začneme psát zdrojový kód, můžeme se setkat s několika barvami, které Visual Basic používá k odlišení:

- modrá - klíčová slova jazyka Visual Basic
- černá - názvy maker, funkčí, proměnných, odkazy na buňky, listy atd.
- zelená - komentáře



Obrázek 1: Části okna VBA editoru

- červená - chybné řádky

Pro komentáře se ve VBA používá apostrof: `'`. Text, který je za apostrofem je při výkonu programu ignorován. Komentáře lze tedy využít i jako nástroj na dočasné odstavení určitých částí makra z běhu.

Zdrojový kód makra tedy může vypadat následovně:

```
Sub Testovaci_makro()
' Komentář
proměnná = 10 ' Do této proměnné byla uložena hodnota 10
End Sub
```

Jelikož není editor VBA počeštěn, je potřeba používat:

- tečku jako oddělovač desetiných míst
- čárku jako oddělovač v seznamu

2.4 Vložení nového makra

Když máme otevřen editor VBA, klikněte dvakrát na `Module_1`, který jsme si vytvořili v předchozím kroku. Každé nové makro musí začínat klíčovým slovem `Sub` a končit klíčovými slovy `End Sub`. Základní kostra nového makra vypadá takto:

```
Sub Moje_nove_makro()
' Zde bude náš kód
End Sub
```

Zdrojový kód makra pak zapisujeme mezi `Sub` a `End Sub`. Přitom je potřeba mít na paměti několik pravidel:

- všechny příkazy piště malými písmemy - pokud je totiž příkaz napsán správně, editor daný příkaz po přesunu na další řád provede změnu prvních písmen na velká - tím získáme potvrzení, že jsme příkaz zadali správně.
- při tvorbě vnořených příkazů je nejlepší napsat jejich začátek a ihned doplnit i konečnou část. Následně můžete pokračovat doplněním dalších příkazů dovnitř prvního příkazu. Předejdete tím chybě, kdy Vám bude chybět ukončení jednoho z příkazů.
- při psaní zdrojového kódu používejte komentáře, pomocí kterých budete zaznamenávat, co má daná část makra vykonávat
- při potřebě dočasně odstavit některou část kódu z běhu, je nejlepší před něj přidat apostrof a tzv. ”zakomentovat” daný řádek

V příkladu uvedeném výše jsme pro pojmenování makra použili název `Moje_nove_makro`. Pro pojmenovávání nových maker existuje několik pravidel:

- v názvech se nerozlišují malá a velká písmena
- mohou obsahovat diakritiku, ale není to preferováno
- název musí být v rámci jednoho modulu jedinečný; v jednom modulu nelze vytvořit makro i funkci se stejným názvem

Jako ukázku jednoduchého makra necháme vyvolat tzv. message box, tedy okno se zprávou. Makro bude tedy vypadat takto:

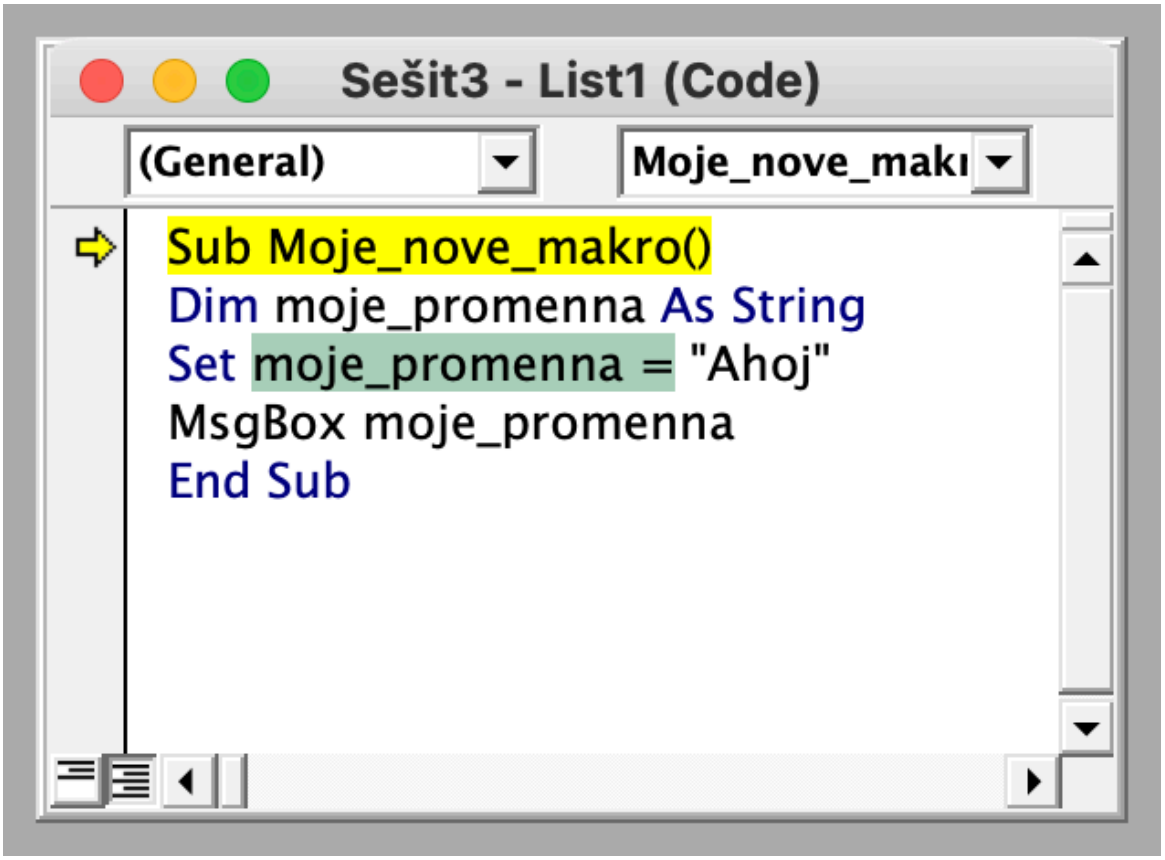
```
Sub Moje_nove_makro()
MsgBox "Ahoj světe!"
End Sub
```

Pokud chceme dané makro spustit, ujistěte se, že máte kurzor položený dovnitř makra `Moje_nove_makro()` a stiskněte klávesu **F5**. Mělo by se Vám zobrazit okno s hlášením ”Ahoj světe!”. Rozeberme si nyní příkaz, který jsme v makru použili.

Příkaz `MsgBox` spouští subproceduru, která je interně definována v základu programovacího jazyka VBA. Subprocedura zobrazuje okno se zprávou. Tato zpráva je subproceduře předávána pomocí parametru - v našem případě jde o parametr `"Ahoj světe!"`, tedy řetězec. Parametr může mít ale i tvar proměnné, ve které uložena nějaká hodnota či tvar hodnoty samotné. Volání této subprocedury může mít i složitější tvary, které si ukážeme na dalších cvičení. Ovšem pro účely seznámení se se základními prvky programovacího jazyka VBA je toto dostačující.

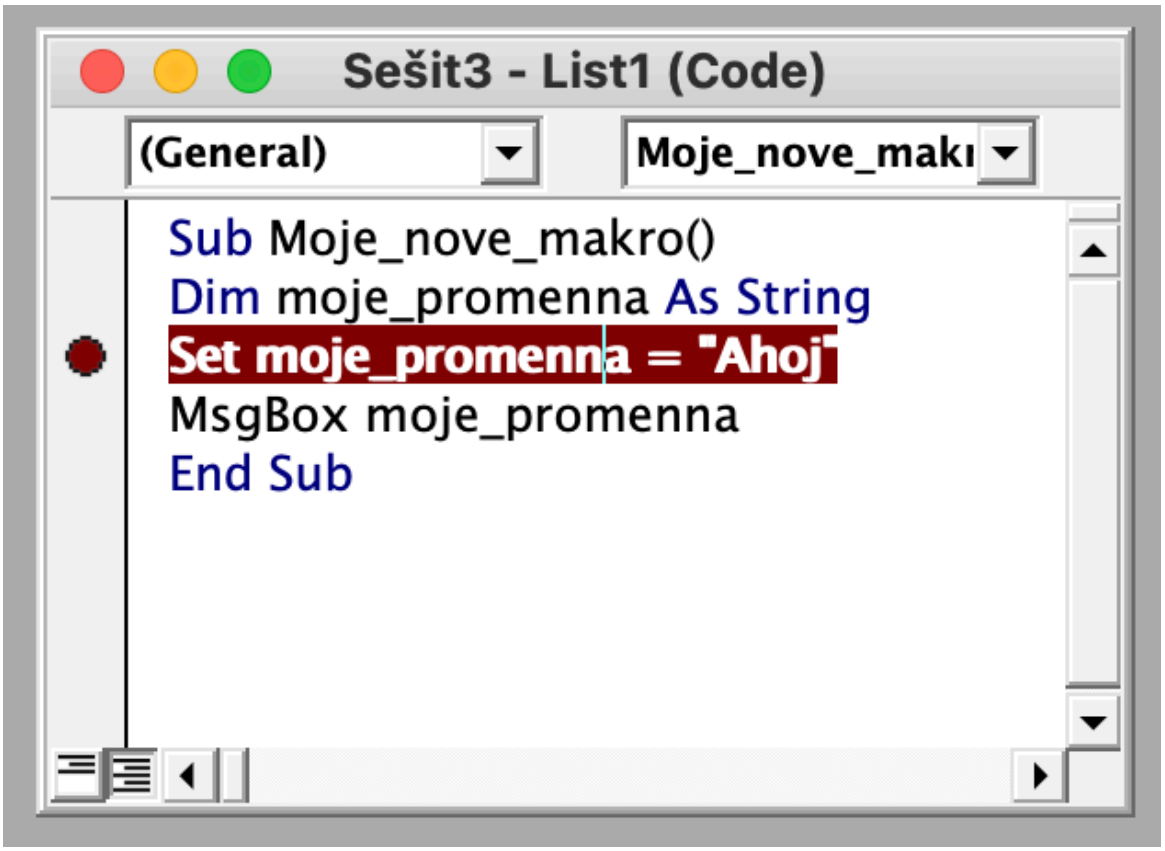
2.5 Ladění při chybě v makru

Ladění lze spustit stiskem klávesy **F8**. Tato klávesa nám umožňuje průběh makra tzv. krokovat. Spuštění tedy probíhá postupně a na každém řádku makra se průběh zastaví a uživatel tak může prozkoumat, jaké jsou v proměnných uložené hodnoty či v případě rozvětvení makra pomocí rozhodovacích struktur (If Else), které části makra jsou či nejsou v průběhu spuštěna.



Obrázek 2: Krokování průběhu makra (pomocí klávesy F8)

Pokud nechceme průběh makra krokovat, je možné na konkrétní řádek nastavit tzv. breakpoint. To je místo, na kterém se průběh programu zastaví bez ohledu na jeho předchozí průběh. V danou chvíli je pak možné stejně jako v předchozím případě nahlížet do obsahu jednotlivých proměnných a identifikovat tak problémovou oblast, která způsobuje nevyžádané jednání makra. Breakpoint se nastaví kliknutím na levou část řádku s kódem. Na obrázku je to přesně místo červené tečky.



Obrázek 3: Nastavení breakpoint

2.6 Uživatelské funkce

Uživatelské funkce se tvoří velmi podobně, jako makra. Místo subprocedury budeme ale vkládat novou Funkci, a proto zápis bude následující:

```
Public Function Moje_funkce()

End Function
```

Všimněme si, že struktura zápisu je podobná zápisu makra. Je zde ovšem klíčové slovo `Public`, o kterém si budeme více říkat později, a klíčové slovo `Function`, které značí, že se nejedná o subproceduru, ale funkci. Tuto funkci pak budeme moci použít při výpočtech v makrech či přímo v buňkách tabulek v Excelu.

Funkce má většinou jeden či více vstupních parametrů, které je potřeba nadefinovat. Tyto parametry se zapíšou do závorek v prvním příkazu funkce a oddělí se čárkami. Návratovou hodnotu pak vrátím zavoláním názvu funkce a vložením "=" před návratovou hodnotu. Celá uživatelem definovaná funkce pak může vypadat následovně:

```
Public Function Moje_funkce(hodnota, exponent)
Moje_funkce = hodnota * exponent
End Function
```

Pokud by se nejednalo o povinný parametr, lze použít klauzuli `Optional`. Výsledný zápis by tedy vypadal takto:

```
Public Function Moje_funkce(hodnota, Optional exponent)
Moje_funkce = hodnota * exponent
End Function
```

Funkce by samozřejmě nedávala v tomto tvaru smysl, ale pro demonstraci je to dostačující.