

# MICROSOFT EXCEL 2010

VBA – syntax jazyka

Mgr. Ján Žitniak

## Obsah

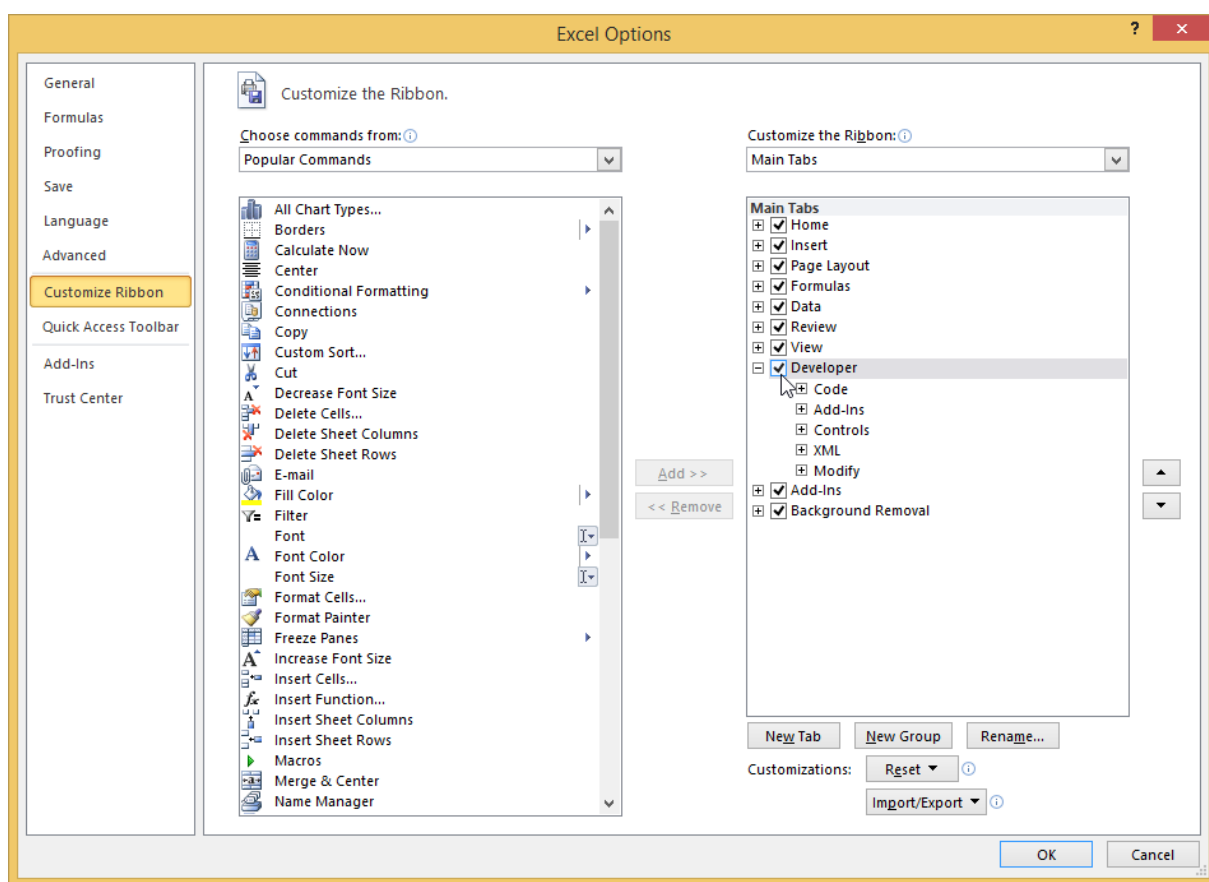
Ako programovať vo Visual Basic for Applications.....	4
Ukladanie súboru .....	5
Povolenie makier .....	5
Prostredie Microsoft Visual Basic.....	7
Programovanie vo VBA.....	7
Formátovanie kódu .....	10
Nahrávanie makra vs programovanie – efektívnosť kódu .....	11
Tvorba vlastnej funkcie (function).....	11
Doplnenie pomocníka k používateľským funkciám .....	13
Čo je premenná .....	13
Dátové typy .....	15
Zrovnanie dátového typu VBA s formátom bunky v MS Excel .....	15
Vypisovanie hodnôt (MsgBox, Debug.print) .....	16
Príklad .....	17
Vstup hodnoty prostredníctvom okna .....	18
Najpoužívanéjšie funkcie (textové, matematické, logické, dátumové a cyklické) .....	19
Matematické operácie s číslami .....	26
Spájanie reťazcov .....	26
With...End With .....	26
Komentáre .....	27
Objekty .....	27
Ukážka kódov – príkladov na objekty.....	28
Vybraté akcie s bunkou .....	28
Formátovanie buniek .....	29
Práca s formulárom .....	30
Dodatočné informácie.....	32
Zabezpečenie makra pred jeho úpravou heslom .....	32
Export makra .....	33
Import makra.....	33
Poznámky .....	33
Ako označiť celú tabuľku .....	33
Ako vložiť vzorec do aktuálnej a definovanej bunky .....	33
Relatívne adresovanie .....	33

Ako premenovať graf.....	34
Praktické odkazy.....	34
Úlohy .....	34

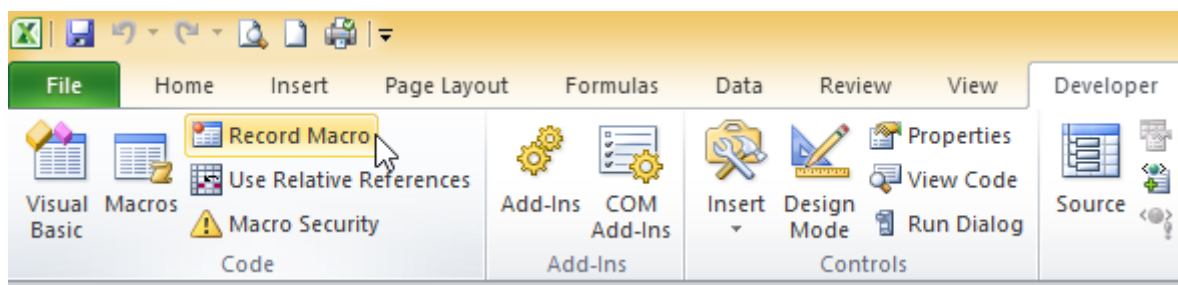
## Ako programovať vo Visual Basic for Applications

V niektorých prípadoch potrebujeme používať funkčnosť mimo rozsah tej, ktorú ponúka Microsoft Excel. Určitým spôsobom „zachraňuje situáciu“ nahraté makro, ktoré je však primárne určené na zautomatizovanie činností v Exceli. Špecifickú funkčnosť si môžeme naprogramovať a to prostredníctvom **Microsoft Visual Basic for Applications**. Microsoft Visual Basic for Applications je „odľahčená“ verzia programovacieho jazyka Microsoft Visual Basic.

Pre sprístupnenie prostredia **Microsoft Visual Basic for Applications** si musíme najprv zobraziť kartu **Developer** a to cez **File > Options**. V otvorenom okne Excel Options vľavo zvolíme možnosť **Customize Ribbon** a úplne vpravo začiarkneme **Developer**.

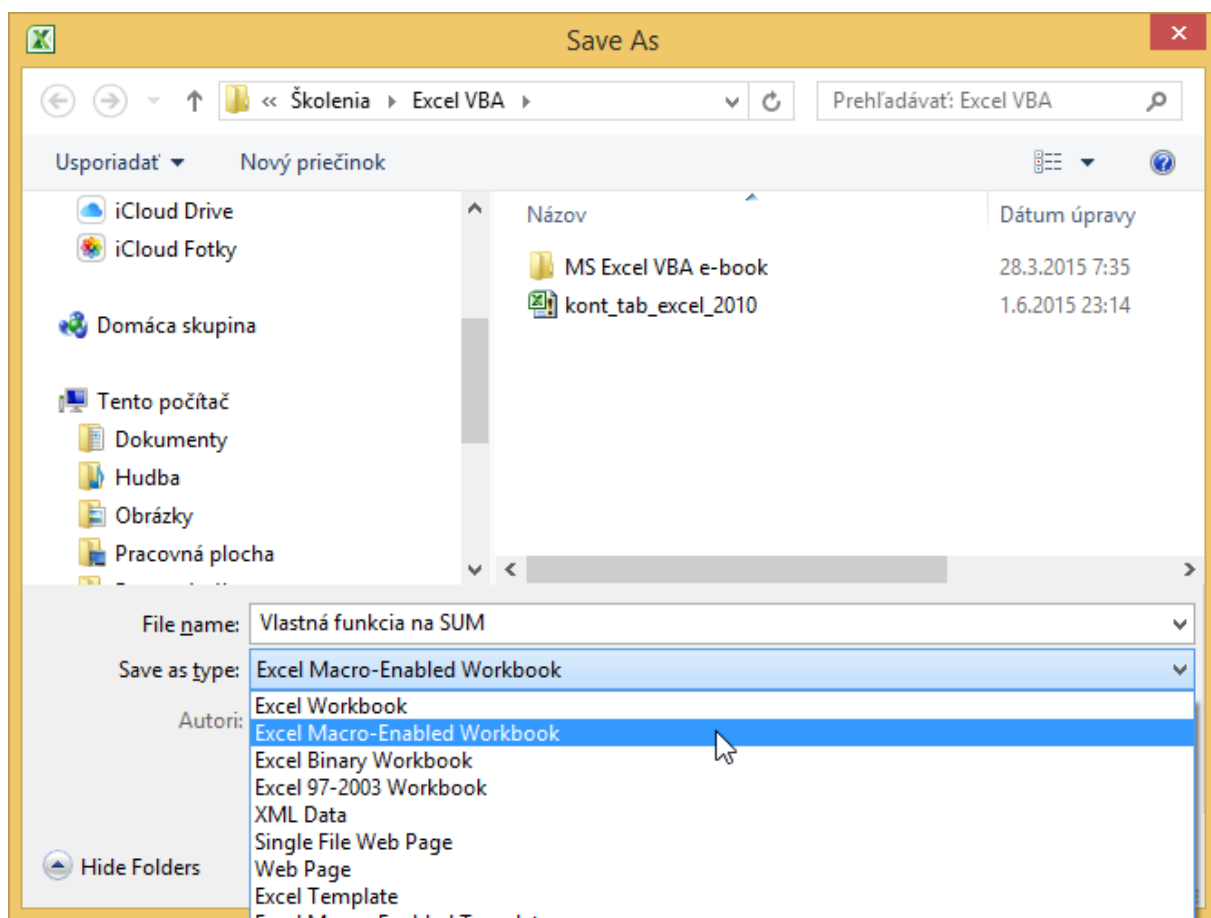


Karta **Developer** sa zvyčajne umiestni ako posledná karta medzi ostatnými.



## Ukladanie súboru

Narozdiel od starších verzií Microsoft Excel musíme súbor s makrom (či už nahratým alebo naprogramovaným) uložiť pod špeciálnym formátom a to: **Excel Macro-Enabled Workbook**.

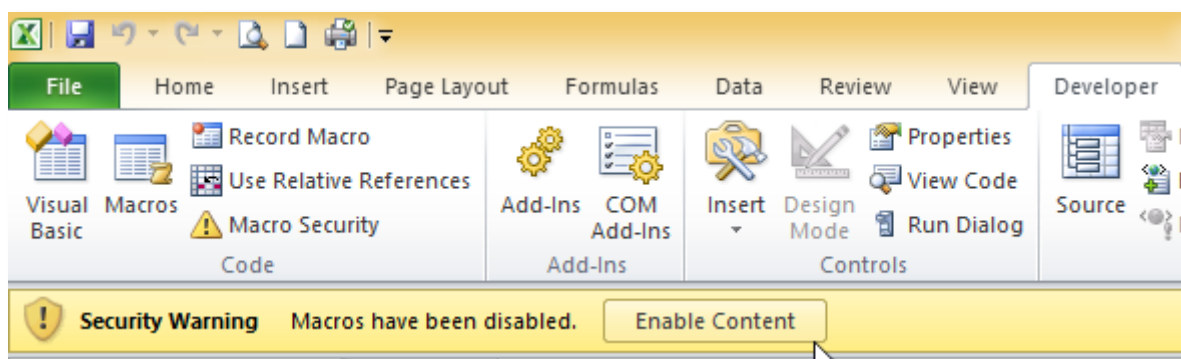


Ikonka súboru vyzerá nasledovne (súbor má príponu .xlsm):

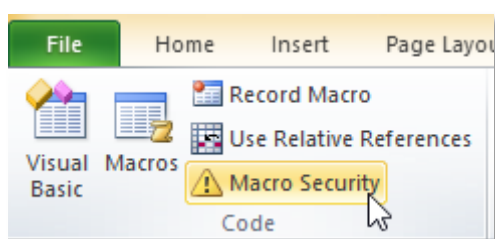


## Povolenie makier

Kvôli bezpečnosti je implicitne povolenie makier zakázané (v minulosti sa prostredníctvom makier spúšťali vírusy). Používateľ musí po otvorení súboru makro daného súboru povoliť.



To ako bude Excel pristupovať k súborom obsahujúce makro závisí od globálneho nastavenia **Macro Security** v karte **Developer**.



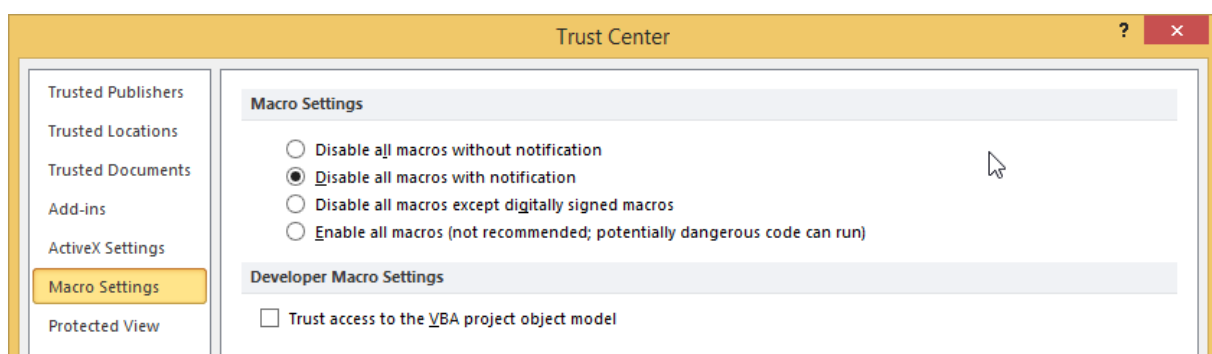
Máme na výber nasledovné možnosti:

**Disable all macros without notification** – zakáže všetky makrá, notifikácia sa nezobrazí

**Disable all macros with notification** – zakáže všetky makrá, notifikácia sa zobrazí

**Disable all macros except digitally signed macros** – zakáže všetky makrá okrem digitálne podpísaných

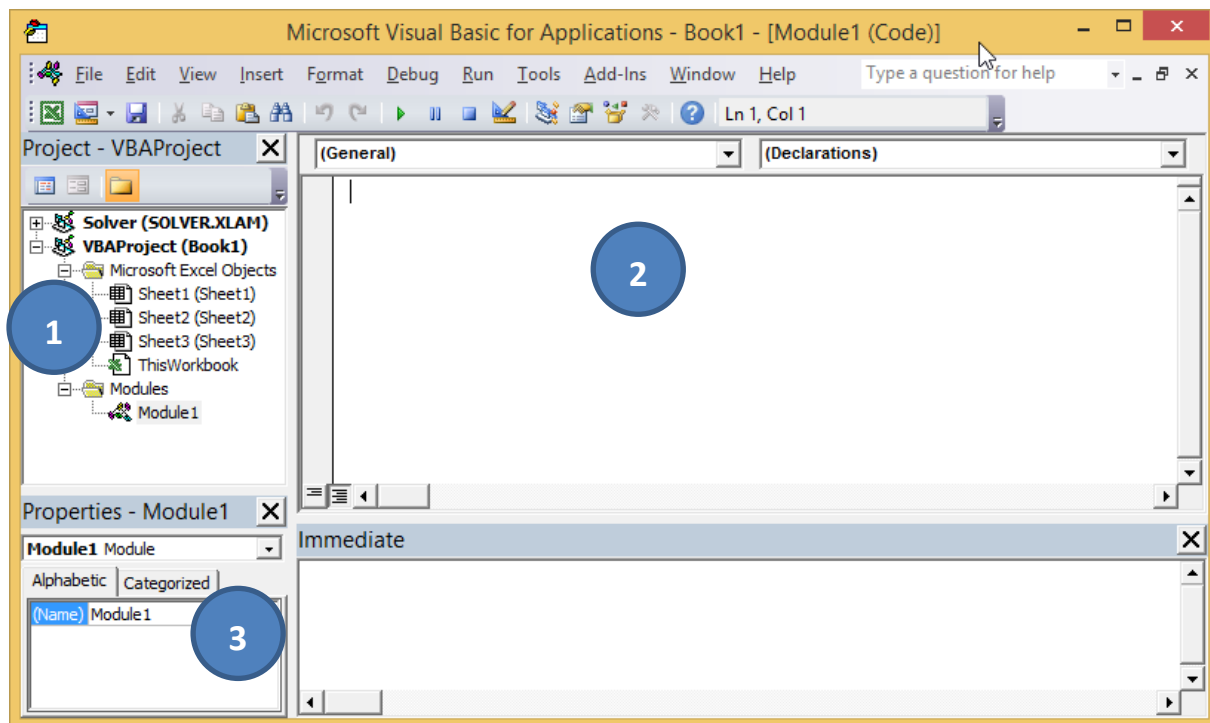
**Enable all macros (not recommended; potentially dangerous code can run)** – povolí všetky makrá (neodporúča sa; potencionálne sa môže spustiť nebezpečný kód)



## Prostredie Microsoft Visual Basic

Karta **Developer > Visual Basic**

Po zvolení tlačidla Visual Basic dostaneme nasledovné okno:



1. Naľavo vidíme tzv. **Project Explorer**, ktorý obsahujú objekty otvoreného zošita (alebo zošitov).
2. Napravo nájdeme **okno editora**, kde budeme písať (alebo po nahraní makra vidieť) samotný kód VBA.
3. Okno, ktoré zobrazuje vlastnosti aktívneho objektu. Využijeme hlavne pri tvorbe formulárov

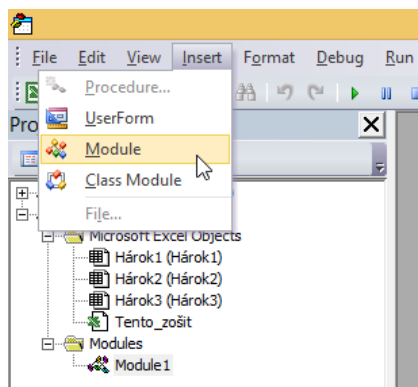
## Programovanie vo VBA

V princípe môžeme programovať dva typy programov

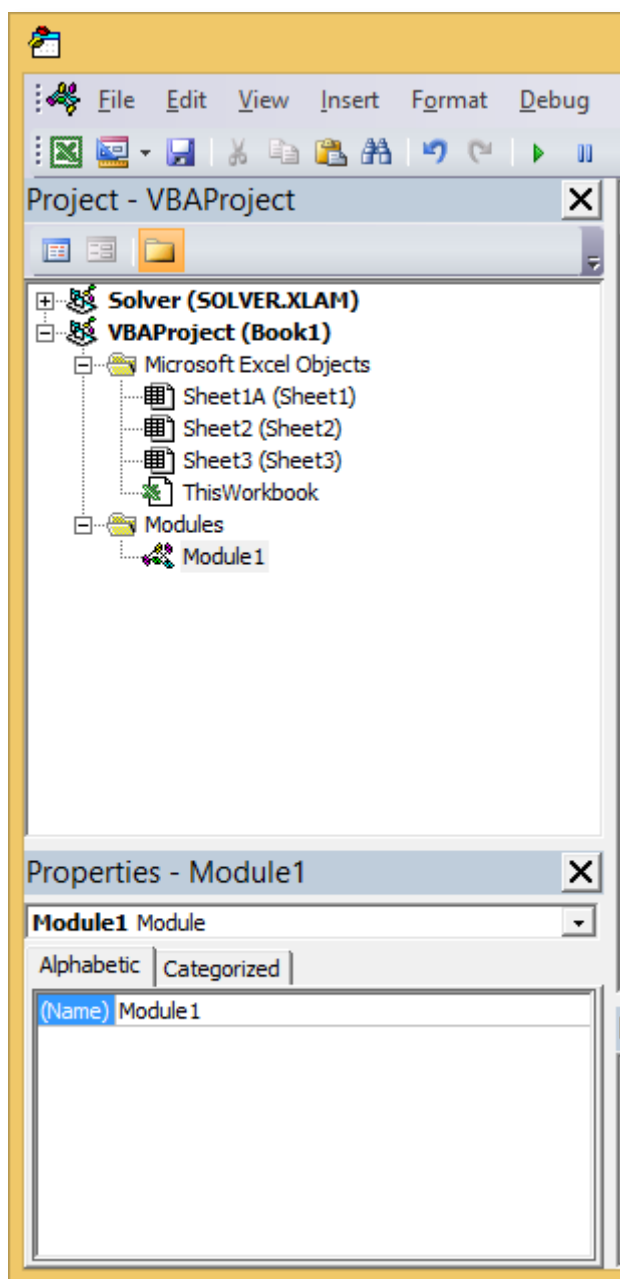
**Sub = Subroutine = podprogram.** Obsahuje príkazy Microsoft VBA, ktoré v Exceli vykonajú konkrétnu činnosť (napr. vymazanie oblasti buniek, zlúčenie, tvorba grafu, kontingenčnej tabuľky, ...). Subroutine nevracia do bunky žiadnu hodnotu.

**Function = funkcia.** . Obsahuje príkazy Microsoft VBA, ktoré využijeme napr. k vytvoreniu vlastnej funkcie (napr. obvod obdĺžnika, vnútorný obvod rúry a podobne). Function vracia návratovú hodnotu.

V oboch prípadoch začneme vložením modulu a to zvolením **Insert a Module**, tak ako to vyjadruje nasledovný obrázok:

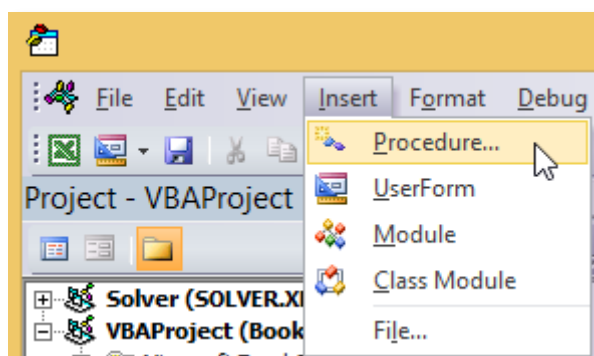


V **Project Explorer** sa zobrazí skupina **Modules**, ktorá obsahuje modul s názvom **Module1**.

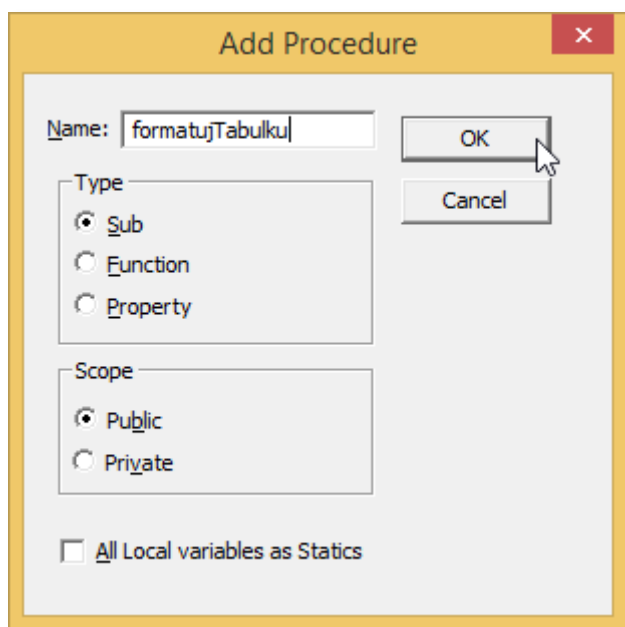


V ďalšom kroku prejdeme do **Insert > Procedure...**





Máme na výber z niekoľkých možností. Podľa situácie vyberieme **Sub** ak chceme vytvoriť podprogram (niekde označovaná aj procedúra) alebo **Function**, ak chceme vytvoriť funkciu. **Property** využijeme na získanie alebo priradenie vlastností objektov.



Po potvrdení môžeme do okna kódu písať zdrojový kód vo Visual Basic for Applications. Formátovanie tabuľky môže vyzeráť nasledovne:

```
Sub Macro1()  
,  
' Macro1 Macro  
,  
  
    ActiveCell.Range("A1:A3").Select  
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone  
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone  
    With Selection.Borders(xlEdgeLeft)  
        .LineStyle = xlContinuous  
        .ColorIndex = 0  
        .TintAndShade = 0  
        .Weight = xlThin  
    End With
```

```

With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
Selection.Borders(xlInsideVertical).LineStyle = xlNone
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
End Sub

```

Jedná sa o tzv. subroutine.

Kopírovanie hárka vyzerá nasledovne:

```

Sub kopirujHarok()
'
' kopirujHarok Macro
'
    Sheets("Sheet1").Select
    Sheets("Sheet1").Copy After:=Sheets(3)
End Sub

```

## Formátovanie kódu

Pri kóde, ktorý zapisujeme do editora by sme sa mali držať nasledovných odporúčaní:

- odsadenie
- príkazy píšeme do jednotlivých riadkov

Nevhodná ukážka formátovania kódu:

```

Sub kopirujHarok()
'
' kopirujHarok Macro
'
Sheets("Sheet1").Select
Sheets("Sheet1").Copy After:=Sheets(3)
End Sub

```

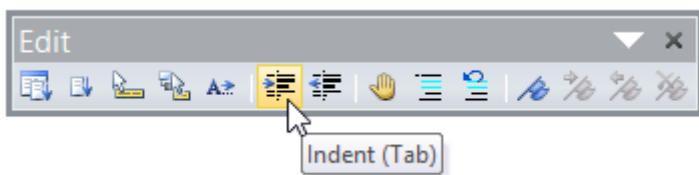
Vhodná ukážka formátovania kódu:

```

Sub kopirujHarok()
'
' kopirujHarok Macro
'
    Sheets("Sheet1").Select
    Sheets("Sheet1").Copy After:=Sheets(3)
End Sub

```

Môžeme použiť panel **Edit** (menu **View > Toolbar > Edit**), kde nájdeme **Intend** (odsadenie doľava) alebo **Outdend** (odsadenie doprava)



## Nahrávanie makra vs programovanie – efektívnosť kódu

Makro, ktoré nahráme má vo väčšine prípadoch zbytočne dlhý kód. Porovnajme nasledovné riešenia: Kód na vyfarbenie jednej bunky (prostredníctvom nahrávania makra):

```

Sub vyfarbi()
'
' vyfarbi Macro
'
'
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
End Sub

```

Ako by sme to mohli naprogramovať my:

```

Sub vyfarbi()
'
' vyfarbi Macro
'
'
    ActiveCell.Interior.Color = 65535
End Sub

```

## Tvorba vlastnej funkcie (function)

Aj keď Microsoft Excel obsahuje veľké množstvo funkcií (SUM, AVERAGE, MIN, MAX, COUNT, IF, ...), niekedy je potrebné v praxi použiť svoju vlastnú (napr. obvod obdĺžnika, vnútorný objem rúry). Na

vytvorenie vlastnej funkcie využijeme makro, ktoré si naprogramujeme v Microsoft Excel VBA. VBA = Visual Basic for Applications.

Ukážku si predvedieme na vytvorenie vlastnej funkcie sčítania dvoch buniek.

V karte **Developer** zvolíme tlačidlo **Visual Basic**.

Do neho zapíšeme nasledovný zdrojový kód:

```
Function obvodObdlznika(stranaA, stranaB)
    obvodObdlznika = 2 * (stranaA + stranaB)
End Function
```

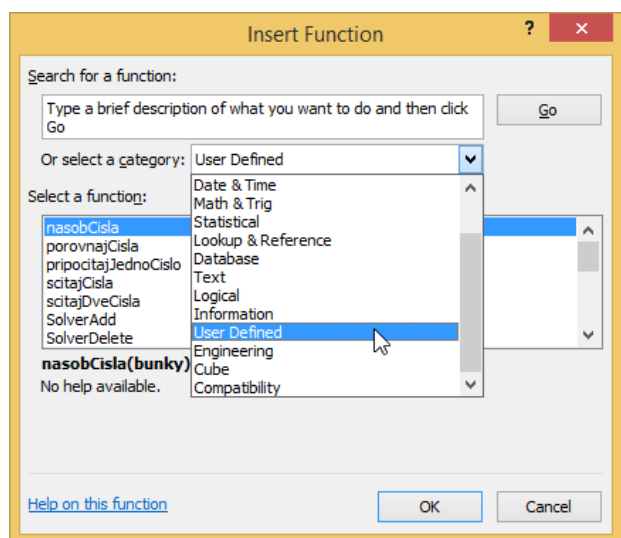
Zdrojový kód obsahuje anglické slová (príkazy, ktorým rozumie Microsoft Excel VBA) a slovenské slová, ktoré sme zadali my.

Function obvodObdlznika(stranaA, stranaB) – je náš vlastný názov funkcie (bez medzier a diakritiky), ktorú potom použijeme v hárku. Obsahuje dva argumenty (premenné): stranaA pre označenie prvej bunky, s ktorou počítame a stranaB pre označenie druhej bunky.

obvodObdlznika = 2 \* (stranaA + stranaB) – Excel VBA vypočíta obvod na základe vzorca, ktorý sme zadali a výsledok „odovzdá“ do premennej s názvom obvodObdlznika. Názov premennej obvodObdlznika musí byť identický s názvom, ktorý sme použili pri príkaze Function obvodObdlznika(stranaA, stranaB)

End Function – týmto príkazom končí funkcia

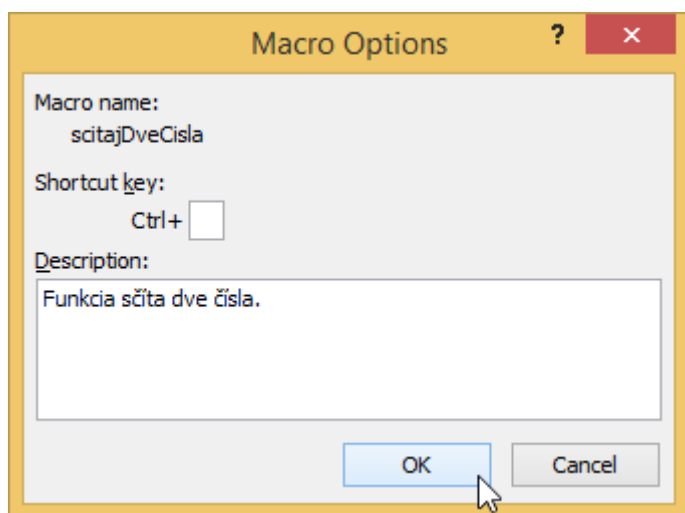
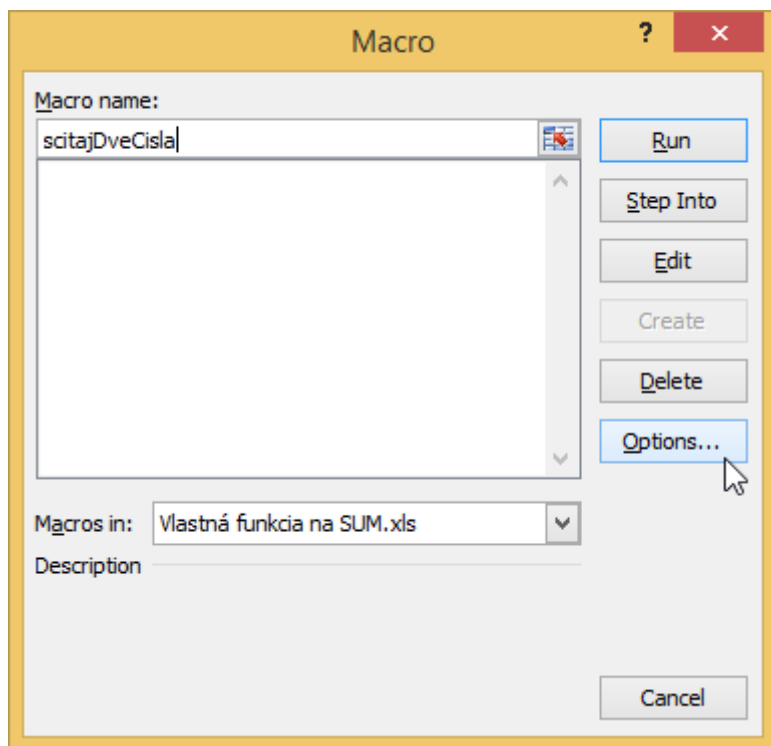
V Exceli vložíme nami definovanú funkciu podobne ako jednu z ďalších existujúcich, viď obrázok nižšie:



## Doplnenie pomocníka k používateľským funkciám

Karta **Developer > Macros**

V karte **Developer** vyberieme tlačidlo **Macros**. Ak sa nami vytvorená funkcia v zozname nenachádza, je potrebné ju vypísať (musíme mať však otvorený súbor, kde sa dané makro nachádza). Ak sa názov funkcie v zozname nachádza, tlačidlá napravo sa zaktívnia. Vyberieme **Options...**



## Čo je premenná

Premenná je dočasné miesto v pamäti, ktorú využijeme na ukladanie ľubovoľnej hodnoty. Každá premenná má názov pozostávajúci z alfanumerických znakov.

Názov premennej by mal logicky nadväzovať na to, čo obsahuje (napr. dph, obrat, vysledok, bunka1, chybovaSprava a podobne). Zápis je potom nasledovný:

```
dph = 1.19
```

```
obratDph = obrat * dph
```

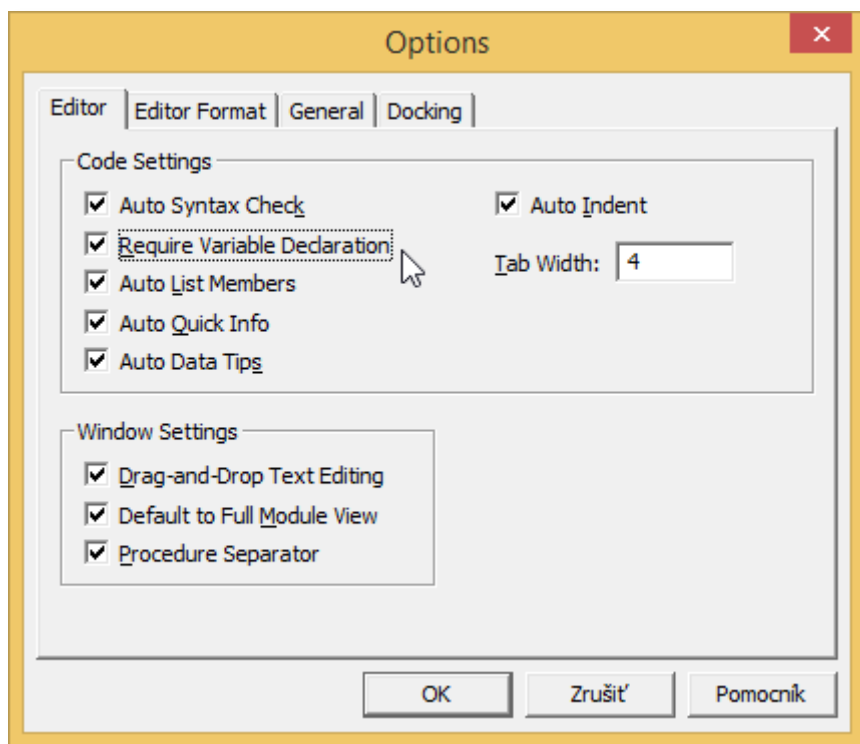
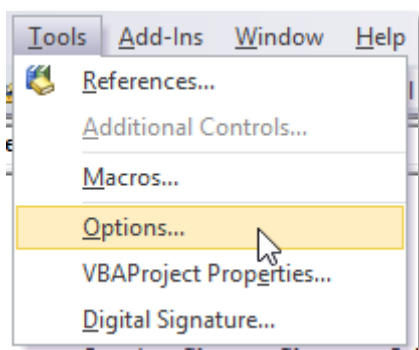
```
chybovaSprava = "Pozor! Zadali ste hodnotu mimo rozsah!"
```

Premennú deklarujeme príkazom **Dim**, napr.

```
Dim obratDph As Double
```

Double je vlastne dátový typ, ktorý reprezentuje desatinné číslo. Premenná by nemala nadobúdať iný typ údajov, inak dôjde k chybe.

Deklarácia typu premennej nie je záväzná, no kvôli čitateľnosti a prehľadnosti kódu odporúčam použiť nastavenie **Require Variable Declaration** v okne **Options**, menu **Tools**.



## Dátové typy

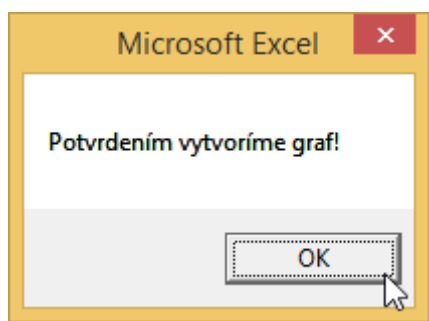
Dátový typ	Počet bytov	Rozsah hodnôt
Boolean	2	True alebo False
Integer	2	-32,768 do 32767
Long	4	-2,147,483,648 do 2,147,483,647
Single	4	-3.402823E38 do 1.401298E45
Double (negative)	8	-1.79769313486232E308 do - 4.94065645841247E-324
Double (positive)	8	4.94065645841247E-324 do 1.79769313486232E308
Currency	8	-922,337,203,685,477.5808 do 922,337,203,685,477.5807
Date	8	1/1/100 to 12/31/9999
String	1(ak 1 znak)	Veľkosť závisí od počtu znakov
Object	4	Akýkoľvek definovaný objekt
Variant	Variabilný	Akýkoľvek dátový typ
Used defined	Variabilný	Variabilný

## Zrovnanie dátového typu VBA s formátom bunky v MS Excel

Visual Basic for Application	Excel
Byte	(nemá)
Boolean	(nemá)
Integer	Číslo
Long	Číslo
Date	Dátum
Object	(nemá)
String	Text
Variant	Všeobecný

## Vypisovanie hodnôt (MsgBox, Debug.print)

Použijeme v prípade, ak chceme používateľovi vypísať v okne správu. Podľa typu odpovede (tlačidla) môžeme ďalej spracovať odpoveď.



Príkaz, ktorý sme použili:

```
MsgBox "Potvrdením vytvoríme graf!"
```

Syntax príkazu

```
MsgBox(prompt, buttons , title, helpfile, context)
```

**prompt** – povinný parameter. Je vlastne reťazec, ktorý sa v okne zobrazí

**buttons** – nepovinný parameter. Obsahuje číslo typu tlačidla podľa ktorého sa tlačidlo bude zobrazovať.

Vybrané typy tlačidiel

Constant	Value	Description
<b>vbOKOnly</b>	0	Display <b>OK</b> button only.
<b>vbOKCancel</b>	1	Display <b>OK</b> and <b>Cancel</b> buttons.
<b>vbAbortRetryIgnore</b>	2	Display <b>Abort</b> , <b>Retry</b> , and <b>Ignore</b> buttons.
<b>vbYesNoCancel</b>	3	Display <b>Yes</b> , <b>No</b> , and <b>Cancel</b> buttons.
<b>vbYesNo</b>	4	Display <b>Yes</b> and <b>No</b> buttons.
<b>vbRetryCancel</b>	5	Display <b>Retry</b> and <b>Cancel</b> buttons.
<b>vbCritical</b>	16	Display <b>Critical Message</b> icon.
<b>vbQuestion</b>	32	Display <b>Warning Query</b> icon.
<b>vbExclamation</b>	48	Display <b>Warning Message</b> icon.
<b>vbInformation</b>	64	Display <b>Information Message</b> icon.

**title** – nepovinný parameter, zobrazí okno nami definovaným nadpisom

**helpfile** – nepovinný parameter, obsahuje odkaz na súbor pomocníka

**context** –nepovinný parameter hesla zo súboru pomocníka

Návratová hodnota



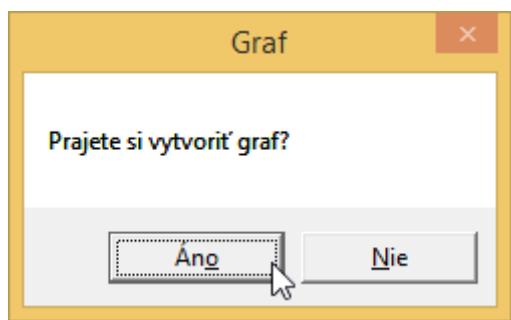
Constant	Value	Description
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

### Príklad

```

Sub msgBoxGraf()
    Dim odpoved As Integer
    Dim otazka As String
    Dim titulok As String
    ' Prompt
    otazka = "Prajete si vytvoriť graf?"
    ' Dialog's Title
    titulok = "Graf"
    'Display MessageBox
    odpoved = MsgBox(otazka, vbYesNo, titulok)
    ' Check pressed button
    If odpoved = vbNo Then
        MsgBox "Odpovedali ste NIE!"
    Else
        MsgBox "Odpovedali ste ÁNO!"
    End If
End Sub

```



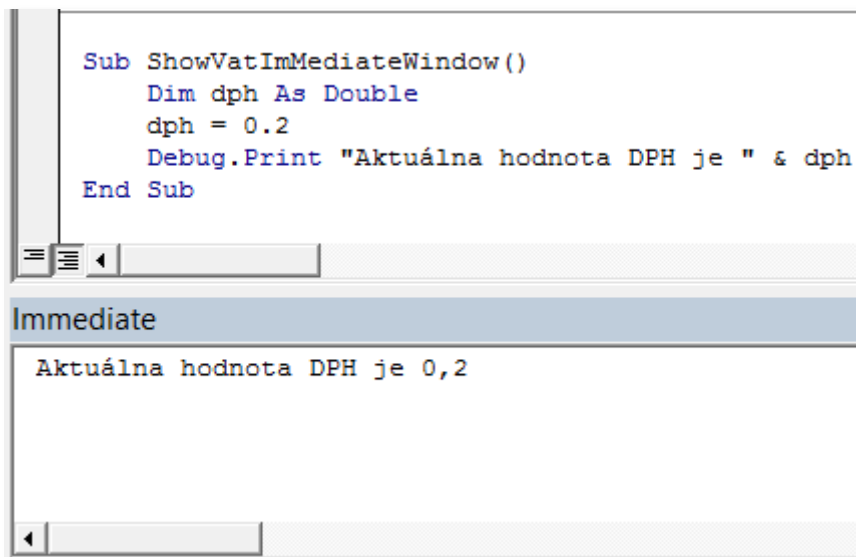
Na vypisovanie hodnôt môže použiť príkaz:

```
Debug.Print
```

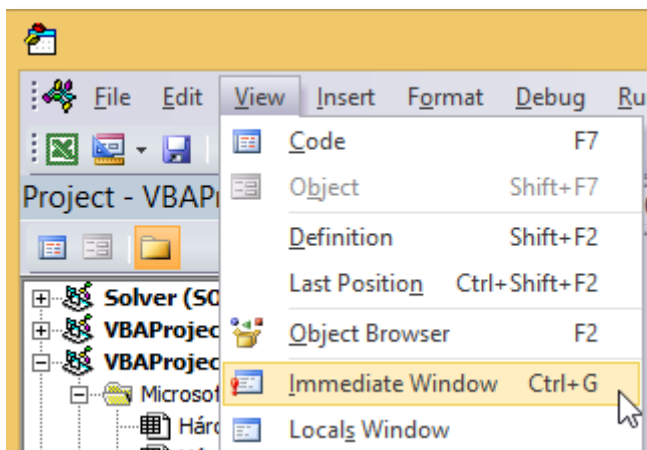
Príklad:

```
Debug.Print "Aktuálna hodnota DPH je " & dph
```

Narozdiel do MsgBox, výstupnú hodnotu zobrazí v **Immediate Window**

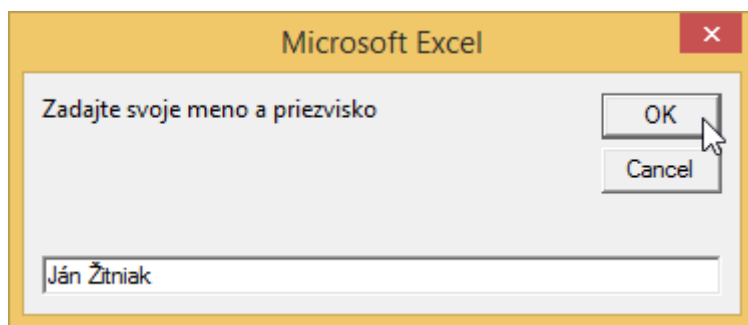


Poznámka: Immediate Window môžeme zobrazíť/skryť cez **View > Immediate Window**)



## Vstup hodnoty prostredníctvom okna

Pri MsgBox sme smerom k používateľovi vypísali nejakú správu, no v niektorých prípadoch potrebujeme, aby používateľ zadal hodnotu, ktorú chceme dodatočne spracovať



Príkaz, ktorý sme použili:

```
InputBox "Zadajte svoje meno a priezvisko"
```

Syntax príkazu

```
InputBox(prompt, title, default, left, top, helpfile, helpcontextid, type)
```

najdôležitejšie parametre

**prompt** – povinný parameter. Je vlastne reťazec, ktorý sa v okne zobrazí

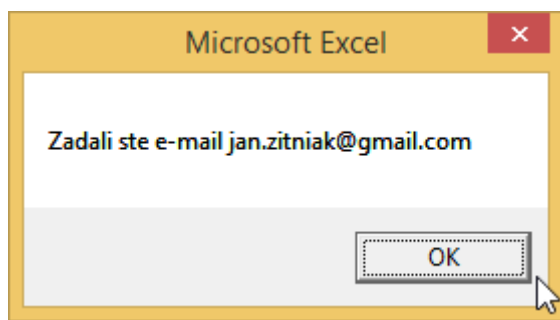
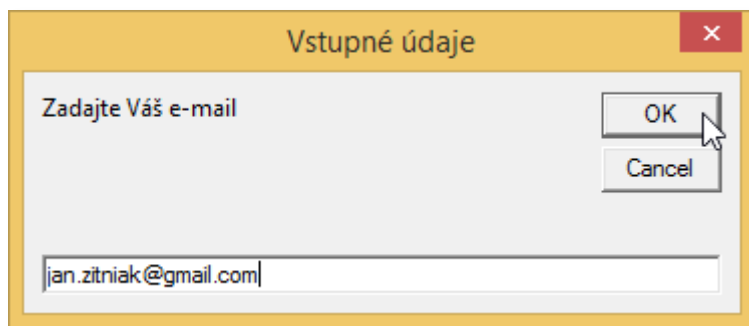
**title** – nepovinný parameter. Titulok okna

**default** – východzia hodnota, ktorú vopred Excel vypíše

Odpoveď môžeme spracovať napr. takýmto spôsobom

```
Sub ZadaJEmail()  
    Dim odpoved As String  
    odpoved = InputBox("Zadajte Váš e-mail", "Vstupné údaje", "@")  
    MsgBox ("Zadali ste e-mail " & odpoved)  
End Sub
```

Ukážka použitia a spracovania InputBox



## Najpoužívanéjšie funkcie (textové, matematické, logické, dátumové a cyklické)

Excel VBA obsahuje množstvo funkcií, my si vypíšeme niektoré z nich.

Zdroj: <http://www.excelfunctions.net/VBA-Operators-And-Functions.html>

Funkcia	Akcia	Kategória funkcií
<b>Abs</b>	Vráti absolútnu hodnotu z čísla. Príklad: <b>Abs(-10)</b> vráti 10 <b>Abs (10)</b> vráti 10	Matematické
<b>Case</b>	Nahrádza použitie funkcie IF (vid' If). Funkciu zapisujeme cez: <b>Select Case (podmienka)</b> Jednotlivé podmienky vypisujeme cez <b>Case Is</b> (ekvivalent <b>Else If</b> ). Ak neplatí žiadna z nich, tak platí podmienka uvedená v <b>Case Else</b> Príklad: <pre>Function AkeCisloCase(cislo)     Select Case cislo         Case Is &lt; 0             AkeCisloCase = "Záporné"         Case Is &gt; 0             AkeCisloCase = "Kladné"         Case Else             AkeCisloCase = "Nezáporné"     End Select End Function</pre>	Logické
<b>Chr</b>	Vráti príslušný znak podľa hodnoty v ASCII tabuľke. Vhodné pre zobrazenie špeciálnych znakov (€, ö, © a podobne). Príklad <b>Char(10)</b> vráti nový riadok <b>Char(97)</b> vráti znak „a“ Zdroj znakov ASCII <a href="http://www.asciitable.com/">http://www.asciitable.com/</a>	Textové
<b>Date</b>	Vráti aktuálny dátum	Dátumové
<b>DateAdd d</b>	Pripočíta špecifický čas k dátumu. Formát funkcie je: <b>DateAdd(Interval, Number, Date),</b> <b>interval</b> – typ argumentu <b>number</b> – počet (hodín, dní, rokov, ...) o koľko pripočíta k dátumu <b>date</b> – dátum, ku ktorému sa pripočítava	Dátumové

#### Legenda ku **interval**

Interval	Reprezentuje
<b>yyyy</b>	Year (rok)
<b>q</b>	Quarter (štvrtrok)
<b>m</b>	Month (mesiac)
<b>y</b>	day of the year (deň roku)
<b>d</b>	Day (deň)
<b>w</b>	Weekday (pracovný deň)
<b>ww</b>	Week (týždeň)
<b>h</b>	Hour (hodina)
<b>n</b>	Minute (minúta)
<b>s</b>	Second (sekunda)

Príklad:

**DateAdd("d", 1, "1.6.2015")** - pridá 1 deň k dátumu 1.6.2015

**DateAdd("ww", 2, "1.6.2015")** - pridá 2 týždne k dátumu 1.6.2015

#### **DateDiff** **f**

Vypočíta počet (interval) medzi dátumami. Formát funkcie je:

**DateDiff(Interval, Date, Date),**

**interval** – typ argumentu (viď tabuľku vyššie pri **DateAdd**)

**date1** – dátum

**date2** – dátum

Príklad:

**DateDiff("d", "1.6.2015", "1.7.2015")** - zistí počet dní medzi 1.6.2015 a 1.7.2015

**DateDiff("w", "1.6.2015", "1.7.2015")** - zistí počet víkendov medzi 1.6.2015 a 1.7.2015

Dátumové

#### **Day**

Vráti deň z dátumu.

Príklad:

**Day ("28.7.2015")** - vráti 28

Dátumové

<p><b>For</b></p>	<p>Vhodne pri tvorbe opakujúceho sa príkazu/príkazov</p> <p>Príklad:</p> <pre>Sub VyplnRiadky()     Dim riadok As Integer     For riadok = 1 To 6         Cells(riadok, 1).Value = 100     Next riadok End Sub</pre> <p>Výsledok: Vyplní prvých šesť riadkov v prvom stĺpci (teda od A1 do A6)</p> <pre>Sub VyplnRiadky()     Dim riadok, aktRiadok, aktStlpec As Integer     aktRiadok = ActiveCell.Row     aktStlpec = ActiveCell.Column     For riadok = aktRiadok To aktRiadok + 6         Debug.Print (riadok)         Cells(riadok, aktStlpec).Value = 100     Next riadok End Sub</pre> <p>Výsledok: Vyplní šesť riadkov vzhľadom na aktuálnu pozíciu označenej bunky</p>	<p>Cyklické</p>
<p><b>Foreach</b></p>	<p>Prechádza kolekciu údajov bez toho, aby vedela koľko ich je (narozdiel od For, kde zadávame konečný počet)</p> <p>Príklad:</p> <pre>Function scitajCisla(bunky As Range)     Dim aktBunka As Range     For Each aktBunka In bunky         scitajCisla = scitajCisla + aktBunka.Value     Next aktBunka End Function</pre> <p>Výsledok: Sčíta označené bunky (ekvivalent SUM)</p>	<p>Cyklické</p>
<p><b>Hour</b></p>	<p>Vráti hodinu z času.</p> <p>Príklad:</p> <p><b>Hour("22:45:00")</b> - vráti 22</p>	<p>Dátumové</p>
<p><b>If</b></p>	<p>Ekvivalent k funkcii IF prakticky s ľubovoľným počtom. Argumentom je podmienka. Ak je podmienka splnená, vráti TRUE, v opačnom prípade FALSE.</p> <p>Poznámka: Príkaz If sa používa v spojení s príkazom:</p> <p><b>Then</b> - ak je podmienka splnená, tak sa vykoná ...</p> <p><b>Else</b> – v opačnom prípade sa vykoná ...</p> <p><b>Elseif</b> – je vlastne vnorené If. Ak neplatí hlavné podmienka, tak ...</p>	<p>Logické</p>

**End If** – koniec používania If

Príklad:

```
Function AkeCislo(cislo)
    If cislo > 0 Then ' dalsi prikaz len do noveho riadka!
        AkeCislo = "Kladné"
    ElseIf cislo < 0 Then ' dalsi prikaz len do noveho riadka!
        AkeCislo = "Záporné"
    Else ' dalsi prikaz len do noveho riadka!
        AkeCislo = "Nezáporné"
    End If
End Function
```

Vypíše odpoveď podľa toho, či je hodnota kladná, záporná alebo nezáporná

Poznámka: Môžeme kombinovať spolu s AND, OR a ďalšími logickými funkciami, napr.:

```
Function AkeCislo(cislo)
    If cislo >= 0 And cislo <= 100 Then
        AkeCislo = "Číslo je v rozsahu od 0 do 100"
    End If
End Function
```

**InStr**

Vracia pozíciu hľadaného reťazca v reťazci. Hľadanie začína na zadanej pozícii.

Textové

Príklad:

**InStr(1, "Here is the searchword", "searchword")** – slovo searchword našiel, preto vráti hodnotu 13

**InStr(14, "Here is the searchword but here is another searchword", "searchword")** - slovo searchword našiel, hľadá však od pozície 14, preto 44

**InStr(1, "www.google.com", ".sk")** - slovo .sk nenašiel, výsledok je 0

**Int**

Vráti celé číslo (nie zaokrúhlené) z desatinného čísla

Matemati  
cké

Príklad:

**Int(5.79)** – vráti hodnotu 5

**IsDate**

Skontroluje, či sa jedná o dátum. Vracia TRUE ak hodnota je dátum, v opačnom prípade FALSE

Dátumové

**IsDate("01/01/2015")** – hodnota JE dátum, preto vráti True

**IsDate(100)** – hodnota NIE JE dátum, preto vráti False

<b>IsError</b>	<p>Skontroluje, či bunka obsahuje chybu. Ak áno, vráti TRUE, v opačnom prípade FALSE.</p> <p>Príklad:</p> <pre>Sub Macro()     Dim cell As range     For Each cell In range("A1:B3")         If IsError(cell) Then             MsgBox ("Error")         End If     Next cell End Sub</pre>	Logické
<b>IsNumeric</b>	<p>Skontroluje či hodnota je číselná alebo nie. Ak áno, vráti TRUE, v opačnom prípade FALSE.</p> <p>Príklad:</p> <pre>Sub CheckIsNumeric()     Dim MyVar, MyCheck     MyVar = "53"      ' Assign value.     MyCheck = IsNumeric(MyVar)    ' Returns True.     MyVar = "459.95"   ' Assign value.     MyCheck = IsNumeric(MyVar)    ' Returns True.     MyVar = "45 Help"   ' Assign value.     MyCheck = IsNumeric(MyVar)    ' Returns False. End Sub</pre>	Logické
<b>Left</b>	<p>Vráti v hľadanom reťazci špecifický počet znakov a to zľava (od začiatku). Formát funkcie je nasledovný:</p> <p><b>Left(String, Length)</b></p> <p>String – reťazec, v ktorom hľadáme</p> <p>Length – počet znakov, ktoré vráti</p> <p>Príklad:</p> <p><b>Left("abcdefghijklm", 4)</b> - vráti reťazec "abcd"</p> <p><b>Left("abcdefghijklm", 1)</b> - vráti reťazec "a"</p>	Textové
<b>Len</b>	<p>Vráti počet znakov reťazca.</p> <p>Príklad:</p> <p><b>Len("abcdefg")</b> - vráti 7, pretože je to 7 znakov</p>	Textové
<b>Month</b>	<p>Vráti mesiac z dátumu.</p> <p>Príklad:</p> <p><b>Month ("28.7.2015")</b> - vráti 7</p>	Dátumové



<b>Mid</b>	<p>Vráti časť textu z textového reťazca na základe pozície odkiaľ a počtu znakov. Formát je nasledovný:</p> <p><b>Mid(String, Start, Length)</b></p> <p>String – textový reťazec, z ktorého vyberieme časť textu</p> <p>Start – štartovacia pozícia, z ktorej vyberieme text</p> <p>Length – počet znakov, ktoré chceme vybrať</p> <p>Príklad:</p> <p><b>Mid("abcdefghijlm", 4, 5)</b> – vráti reťazec "defgh"</p> <p><b>Mid("abcdefghijlm", 10, 2)</b> – vráti reťazec "jk"</p>	Textové
<b>Minute</b>	<p>Vráti minúty z času.</p> <p>Príklad:</p> <p><b>Minute ("22:45:15")</b> - vráti 45</p>	Dátumové
<b>Now</b>	<p>Vráti aktuálny systémový dátum a čas</p>	Dátumové
<b>Right</b>	<p>Vráti v hľadanom reťazci špecifický počet znakov a to zprava (od konca). Formát funkcie je nasledovný:</p> <p><b>Right(String, Length)</b></p> <p>String – reťazec, v ktorom hľadáme</p> <p>Length – počet znakov, ktoré vráti</p> <p>Príklad:</p> <p><b>Right("abcdefghijlm", 4)</b> – vráti reťazec "jklm"</p> <p><b>Right("abcdefghijlm", 1)</b> – vráti reťazec "m"</p>	Textové
<b>Round</b>	<p>Zaokrúhli číslo na konkrétny počet desatinných miest</p> <p>Príklad:</p> <p><b>Round(5.79, 1)</b> – zaokrúhli číslo na 1 desatinné miesto, výsledok je 5,8</p>	Matematické
<b>Second</b>	<p>Vráti sekundy z času.</p> <p>Príklad:</p> <p><b>Second ("22:45:15")</b> - vráti 15</p>	Dátumové
<b>Sqr</b>	<p>Vráti druhú odmocninu z čísla</p> <p>Príklad:</p> <p><b>Sqr(4)</b> – vráti číslo 2</p>	Matematické

<b>Sqr(16)</b> – vráti číslo 4		
<b>Time</b>	Vráti aktuálny systémový čas	Dátumové
<b>Year</b>	Vráti rok z dátumu. Príklad: <b>Month ("28.7.2015")</b> - vráti 2015	Dátumové

## Matematické operácie s číslami

Operátor	Poznámka	Príklad	Výsledok
+	Sčítanie	= 7 + 2	=9
-	Odčítanie	= 7 - 2	=5
*	Násobenie	= 7 * 2	=14
/	Delenie	= 7 / 2	=3,5
\	Celočíselné delenie (vracia celú časť)	= 7 \ 2	=3
Mod	Podiel (vracia zvyšok po delení)	= 7 mod 2	=1
^	Umocňovanie	=2 ^ 3	=8

## Spájanie reťazcov

V Excel VBA nenájde funkciu CONCATENATE tak ako to poznáme v Exceli. Na spájanie reťazcov použijeme znak:

&

Príklad:

```
Sub VypisDPH()
    Dim dph As Double
    Dim sprava As String
    sprava = "Aktuálna DPH je "
    dph = 0.2 * 100
    Debug.Print sprava & dph & "%"
End Sub
```

Vypíše: Aktuálna DPH je 20%

## With...End With

Zefektívňuje zápis zdrojového kódu.

Zvážme nasledovný kód:

```
Public Sub zmenPismo()
    ActiveCell.Font.Bold = True
    ActiveCell.Font.Color = vbBlue
    ActiveCell.Font.Name = "Arial"
    ActiveCell.Font.Size = 22
    ActiveCell.Font.Italic = True
End Sub
```

Rýchlejšie ho môžeme zapísať nasledovný spôsobom:

```
Public Sub zmenPismo()
    With ActiveCell.Font
        .Bold = True
        .Color = vbBlue
        .Name = "Arial"
        .Size = 22
        .Italic = True
    End With
End Sub
```

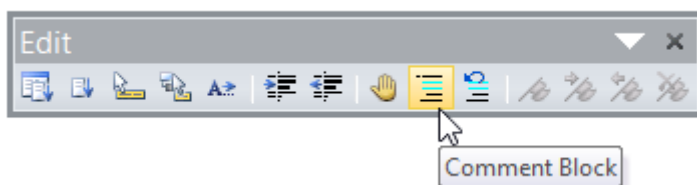
## Komentáre

Použijeme v prípade vkladania poznámky do zdrojového kódu. Slúži na popis funkcionality. Na poznámky použijeme znak:

' – apostrof

```
Sub VypisDPH()
    Dim dph As Double ' Deklaracia premennej typu desatinne cislo
    Dim sprava As String ' Deklaracia premennej typu retazec
    sprava = "Aktuálna DPH je " ' Naplnime premennu textom
    dph = 0.2 * 100 ' Excel VBA nepozna percenta, preto musime pouzit desatinne cislo. To som
    Debug.Print sprava & dph & "%" ' Zlucime vsetky premenne a vypiseme vysledok do konzoly
End Sub
```

Môžeme použiť panel **Edit** (menu **View > Toolbar > Edit**), kde nájdeme **Comment Block** (pridanie komentára) alebo **Uncomment Block** (zrušenie poznámok)



## Objekty

Vo Visual Basicu for Application existuje niekoľko typov objektov:

Application	Aplikácia
Workbook	Zošíť
Worksheet	Hárok (List)

Range	Bunka v hárku alebo oblasť buniek
Cells	Bunka v hárku alebo oblasť buniek - iný formát zápisu
UserForm	Formulár
Module	Modul vo VBA
Class Module	Modul triedy vo VBA

### Ukážka kódov – príkladov na objekty

Názov objektu všeobecne	Praktické použitie	Výstup
Funkce MsgBox - Okno so správou v kóde	MsgBox "Microsoft Excel"	Vypíše okno: "Microsoft Excel"
Workbook – Zožit.xlsx	Workbooks("cogs.xls").Open	Otvorí zožit s názvom "cogs.xls"
Sheets - Hárok.xlsx	Sheets("Hárok1").Select	Označí hárok s názvom Hárok1
Range - Bunka	Range("B3") = "Pondelok"	Do bunky "B3" bude priradená hodnota "Pondelok"
	Range("B3:B10").Select	Vyberie bunky B3 až B10

### Vybraté akcie s bunkou

Sú priradené k objektom alebo prvkom

Akcia	Popis	Prvok	Syntax
AddComment	vlož komentár	bunka	Range("B20").AddComment
Clear	vymaže obsah i formát	bunka	Range("B20").Clear
ClearComments	vymaž komentár	bunka	Range("B20").ClearComments
ClearContents	vymaž obsah	bunka	Range("B20").ClearContents
ClearFormats	vymaž formát	bunka	Range("B20").ClearFormats
Copy	kopíruj	bunka	Range("B20").Copy
Cut	vyjmi	bunka	Range("B20").Cut
Delete	odstráň bunku	bunka	Range("B20").Delete

Find	nájdi	bunka	Range("B20").Find
Formula	vlož vzorec	bunka	Range("B20").Formula
Paste	vlož obsah schránky	bunka	Range("B20").Paste
Select	vyber	bunka	Range("B20").Select
Sort	zoraď	bunka	Range("B20:B25").Sort
SetFocus	prejdi na tento prvok	ComboBox	ComboBox1.SetFocus

## Formátovanie buniek

Ak chceme zafarbiť bunku, tak môžeme použiť:

`ActiveCell.Interior.Color = 36` ' Zafarbi aktualnu bunku bledo oranžovou farbou (viď tabuľka farieb nižšie)

alebo

`Range("A1:A6").Interior.Color = RGB(200, 160, 35)` ` Zafarbi bunky v rozsahu A1 až A6

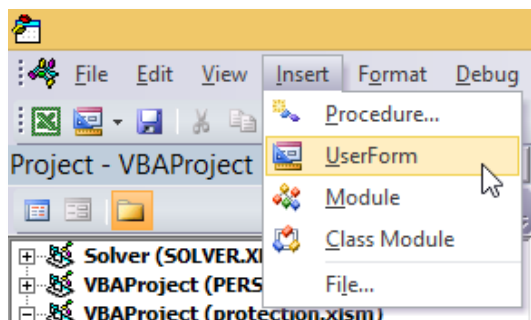
Index	Color	Index	Color	Index	Color
1		21		41	
2		22		42	
3		23		43	
4		24		44	
5		25		45	
6		26		46	
7		27		47	
8		28		48	
9		29		49	
10		30		50	
11		31		51	
12		32		52	
13		33		53	
14		34		54	
15		35		55	
16		36		56	
17		37			
18		38			
19		39			
20		40			

**Poznámka:** Formátovanie buniek neexistuje vo funkcii, iba v makre!

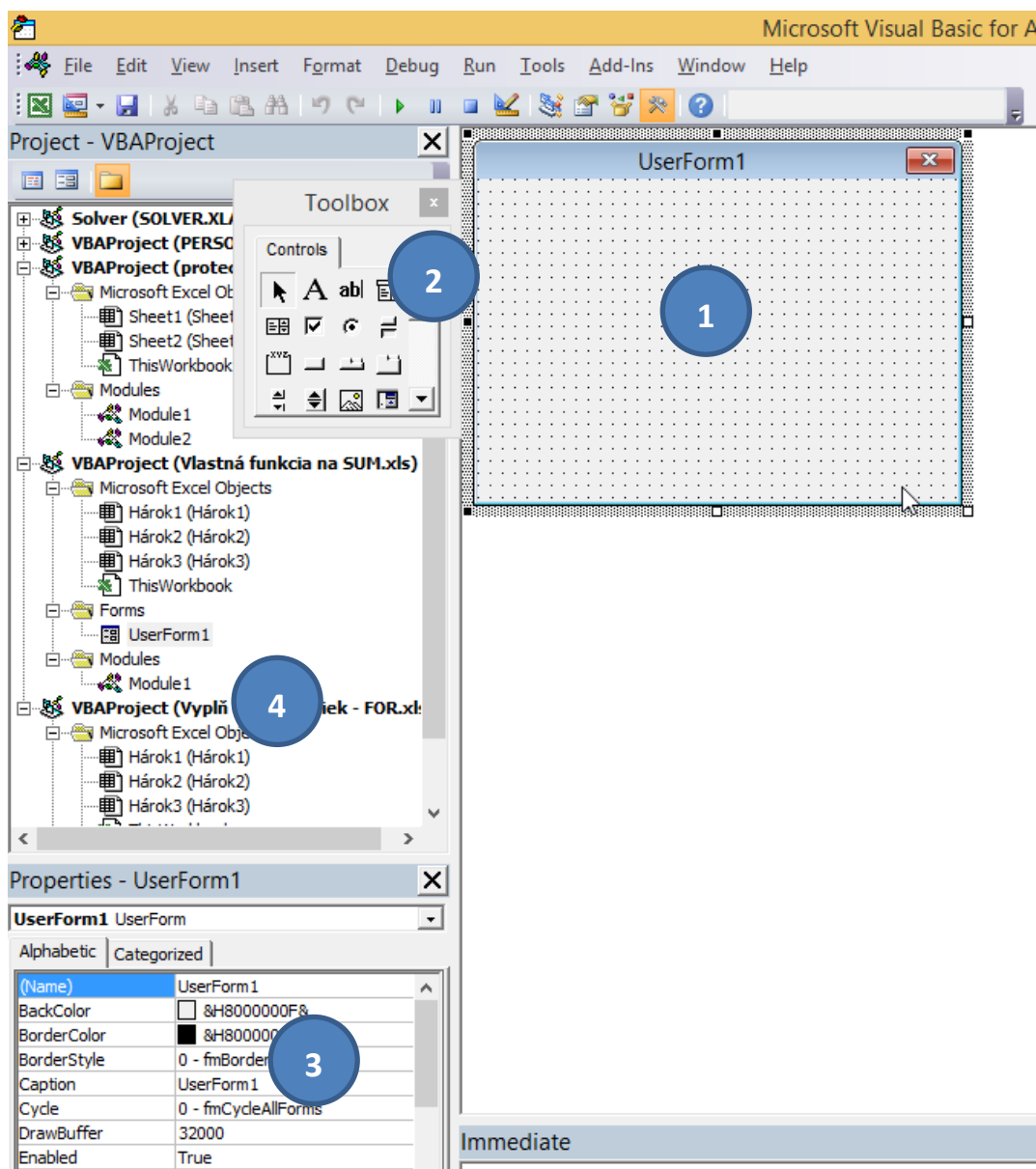
## Práca s formulárom

### Menu Insert > UserForm

Použijeme na tvorbu používateľských formulárov

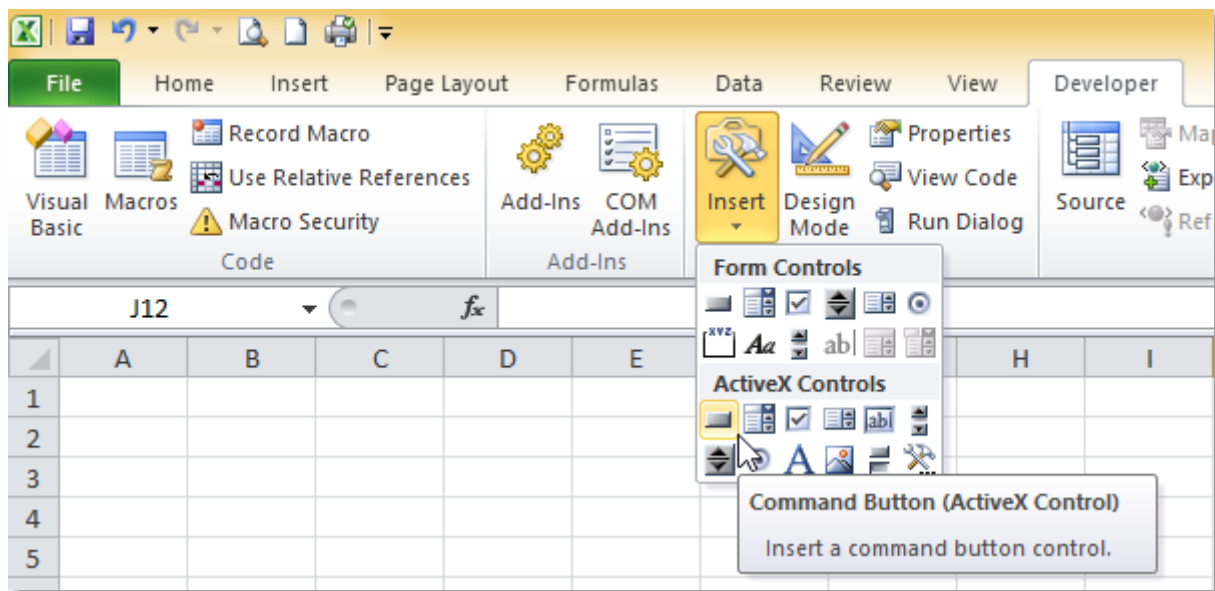


Základom je prázdny formulár, do ktorého postupne pridávame (napr. ťahaním) formulárové prvky (Controls)



1. **Prázdny formulár** – vkladáme formulárové polia
2. **Formulárové polia** – Label, Textbox, Combobox, ListBox, CheckBox, ...
3. **Vlastnosti** – zobrazujú atribúty a vlastnosti jednotlivých častí formulára (napr. Name, Caption, Font, Color a podobne )
4. Vľavo vidíme **názov vytvoreného formulára**

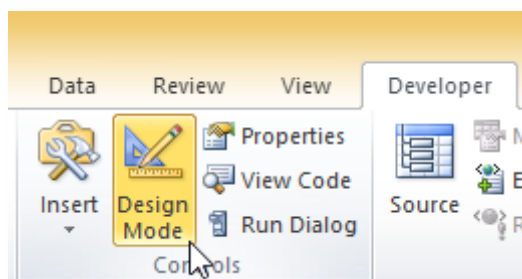
Aby sme formulár z prostredia Excelu mohli otvoriť , vložíme formulárové tlačidlo typu **Command Button** v skupine **ActiveX Controls**.



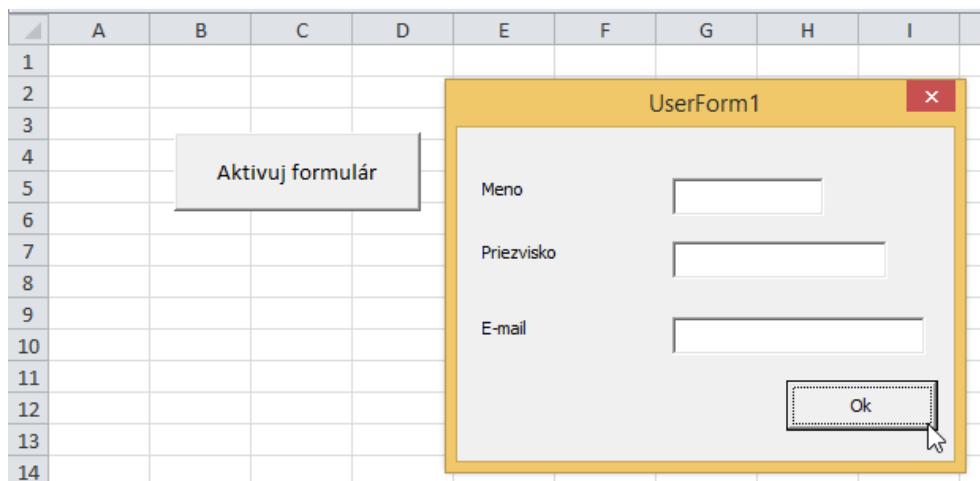
Na vložené tlačidlo klikneme 2x, čím sa dostaneme do prostredia Microsoft Visual Basic for Applications. Dvojklikom sme vlastne vytvorili udalosť (Click), do ktorého zapíšeme nasledovný kód:

```
Private Sub CommandButton1_Click()
    UserForm1.Show
End Sub
```

Zatvoríme okno Excel VBA a priamo v Exceli deaktivujeme úpravu tlačidlom **Design Mode**.



Výsledok:



## Dodatočné informácie

### Dôveryhodné umiestnenia pre makrá:

jednotka\Program Files\Microsoft Office\Templates

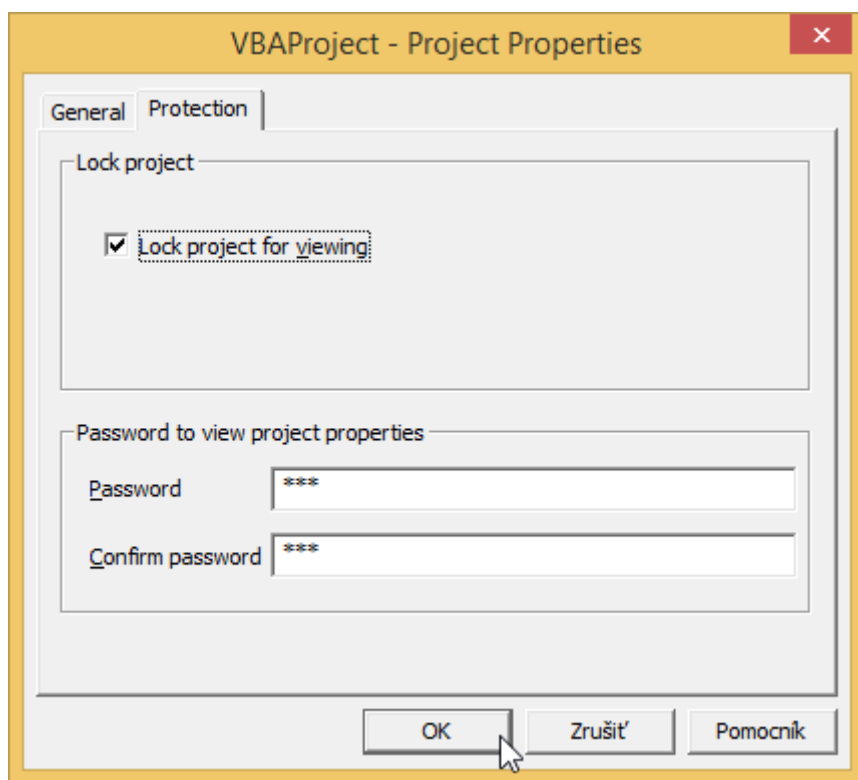
jednotka\Program Files\Microsoft Office\OfficeXY\Startup

## Zabezpečenie makra pred jeho úpravou heslom

V Microsoft Visual Basic IDE je potrebné prejsť do menu:

Tools -> VBAProject properties -> Protection a tam:

Lock project for viewing (po potvrdení kódu je vhodné zavrieť a znovu otvoriť zošit)





## Export makra

V Microsoft Visual Basic IDE je potrebné prejsť do menu:

File -> Export file (uloží súbor s príponou .bas). Jedná sa o textový súbor

## Import makra

V Microsoft Visual Basic IDE je potrebné prejsť do menu:

File -> Import file (nájsť súbor s príponou .bas) . Jedná sa o textový súbor

## Poznámky

Ako označiť celú tabuľku

```
Range("A1").CurrentRegion.Select
```

Ako vložiť vzorec do aktuálnej a definovanej bunky

```
Sub vzorec()  
'  
' vzorec Macro  
'  
    ActiveCell = "=A1+A2" ' vysledok vloží do aktualne oznacenej bunky  
    Range("H3") = "=SUM(B1:B3)" ' vysledok vloží do bunky H3  
End Sub
```

## Relatívne adresovanie

Zvážme nasledovný príklad

```
Sub vzorec()  
'  
' vzorec Macro  
' scita bunku o dva riadky hore a jeden stlpec vľavo + bunku o riadok hore a jeden stlpec vľavo  
    ActiveCell.FormulaR1C1 = "=R[-2]C[-1]+R[-1]C[-1]"  
End Sub
```

Príklad vlastne spočíta obsah bunky o dva riadky hore a jeden stĺpec vľavo + bunku o riadok hore a jeden stĺpec vľavo

Syntax: **RC – row, column (riadok, stĺpec)**

Absolútne adresovanie	Vysvetlenie
<b>R1C1</b>	A1
<b>R3C3</b>	C3
Relatívne adresovanie	
<b>R1C</b>	Prvý riadok, v tom istom stĺpci
<b>RC1</b>	Ten istý riadok, prvý stĺpec
<b>RC[-2]</b>	Pozícia dve bunky vľavo od aktuálneho riadka a stĺpca
<b>RC[+3]</b>	Pozícia tri bunky vpravo od aktuálneho riadka a stĺpca
<b>R[-2]C[-1]</b>	Pozícia dva riadky hore a jeden stĺpec vľavo od aktuálne označenej bunky

## Ako premenovať graf

```
Sub vytvorGraf()  
,  
,  
,  
,  
,  
  
    ActiveSheet.Shapes.AddChart.Select  
    ActiveChart.ChartType = xlColumnClustered  
    ActiveChart.SetSourceData Source:=Range("'stĺpcový graf'!$B$4:$E$11")  
    ActiveChart.Parent.Name = "graf"  
  
End Sub
```

## Praktické odkazy

[www.stackoverflow.com](http://www.stackoverflow.com) – výborné fórum

[www.mrexcel.com](http://www.mrexcel.com) – fórum zamerané na Microsoft Excel

<http://www.excel-easy.com/> - postupy ako pracovať s MS Excel vysvetlené jednoduchým postupom

<http://www.techflutter.com/functions.php> - praktické ukážky použitia jednotlivých funkcií v MS Excel  
VBA aj na stiahnutie

<https://msdn.microsoft.com/en-us/library/office/ee861528.aspx> - referencia na oficiálny web spoločnosti Microsoft ohľadom príkazov v Excel VBA

## Úlohy

Zadania:

- Spravte makro na priemer napr. troch hodnôt
- Výpočet obsahu obdĺžnika
- Vypočítajte obvod obdĺžnika
- Vytvorte výpočet DPH nad nejakou číselnou hodnotou.