

Jak víme, program vykonává řádky kódu postupně, proto není možné výměnu obsahu dvou proměnných realizovat „z ruky do ruky“, tj. stylem  $A = B$ ,  $B = A$ . Po přiřazení  $A = B$  bychom přišli o původní obsah proměnné A. Některé programovací jazyky mají pro tyto účely funkci Swap. Ve VBA si ovšem musíme pomoci jinak.

Univerzální technikou je použití pomocné proměnné.

1	<b>Sub</b> ZamenaObsahuPromennychA()
2	
3	'univerzální postup
4	
5	<b>Dim</b> A <b>As Integer</b>
6	<b>Dim</b> B <b>As Integer</b>
7	
8	'pomocná proměnná
9	<b>Dim</b> C <b>As Integer</b>
10	
11	A = 3
12	B = 5
13	
14	C = A
15	A = B
16	B = C
17	
18	<b>End Sub</b>

Za vyřešení číselné úlohy výše uvedeným způsobem by student získal bod. Další by obdržel, pokud by si dokázal poradit bez pomocné proměnné. To lze provést dokonce dvěma způsoby.

1	<b>Sub</b> ZamenaObsahuPromennychB()
2	
3	'čísla
4	
5	<b>Dim</b> A <b>As Integer</b>
6	<b>Dim</b> B <b>As Integer</b>
7	
8	A = 3
9	B = 5
10	
11	A = A + B
12	B = A - B
13	A = A - B
14	
15	<b>End Sub</b>

1	<b>Sub</b> ZamenaObsahuPromennychC()
2	
3	'čísla
4	
5	<b>Dim</b> A <b>As Integer</b>
6	<b>Dim</b> B <b>As Integer</b>
7	
8	A = 3    '011 ve dvojkové soustavě
9	B = 5    '101 ve dvojkové soustavě
10	
11	'XOR ... exclusive or, exkluzivní disjunkce
12	've VBA operátor Xor
13	
14	'B   A   A Xor B
15	'0   0   0
16	'0   1   1
17	'1   0   1
18	'1   1   0
19	
20	'6 ... 110 ve dvojkové soustavě
21	A = A <b>Xor</b> B
22	'3 ... 011 ve dvojkové soustavě
23	B = A <b>Xor</b> B
24	'5 ... 101 ve dvojkové soustavě
25	A = A <b>Xor</b> B
26	
27	<b>End Sub</b>

Existuje alternativní metoda i pro řetězce?

Ano, a dokonce může být několikrát rychlejší než při užití pomocné proměnné, jež vyžaduje spousty práce s pamětí počítače. Bavíme-li se o pojmu „string“ ve VBA (Visual Basic 4 a novější), pak máme na mysli Basic String (BSTR). Textový řetězec je v paměti uložen ve dvoubajtovém Unicode poli znaků, jemuž předchází 4 bajty (informace o délce), a který je ukončen dvoubajtovým znakem Chr(0). Na místo v paměti přitom míří ukazatel (pointer). Vtip je tedy v tom, že nemusíme měnit přímo obsah dvou proměnných, ale prohodíme jejich ukazatele, podobně jako nějaký rošťák vymění cedulky WC muži a WC ženy. K tomu všemu už ovšem potřebujeme API, především funkci CopyMemory. Upozorňuji, že její nesprávné užití může sestřelit stávající instanci aplikace Excel (nic víc, nic míň, sešit prostě před experimentováním ukládejte). Uvedená deklarace API funkce CopyMemory je platná pro 32bitový Excel. Její syntaxe odpovídá standardu, nicméně si ji řada programátorů přizpůsobuje.

1	<b>Private Declare Sub</b> CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (pDst <b>As</b> _
2	Any, pSrc <b>As</b> Any, <b>ByVal</b> ByteLen <b>As Long</b> )

```

1  Sub ZamenaObsahuTextovychPromennychAPI()
2
3  'datový typ String
4  'práce s ukazateli a API
5
6  'rychlejší oproti pomocné proměnné
7  'dojde k pouhé výměně ukazatelů na řetězce
8
9  Dim X As String
10 Dim Y As String
11
12 Dim T As Long
13
14 X = "Spejbl"
15 Y = "Hurvínek"
16
17 'uložení ukazatele na X
18 T = StrPtr(X)
19 'kopie ukazatele Y do ukazatele X, 32bit
20 CopyMemory ByVal VarPtr(X), ByVal VarPtr(Y), 4
21 'kopie ukazatele T do ukazatele Y
22 CopyMemory ByVal VarPtr(Y), T, 4
23
24 End Sub

```

Funkce StrPtr (string pointer), VarPtr (variable pointer), případně ObjPtr (object pointer) v dokumentaci VBA (VB) jen tak nenajdete. Jsou nedokumentované. Zjednodušeně řečeno, vrací adresu proměnné v paměti.

Pokud se vrátíme k číslům, pak může zpracování čísel typu Long s pomocí API vypadat následovně:

```

1  Sub ZamenaObsahuPromennychLongAPI()
2
3  'datový typ Long
4  'práce s ukazateli a API
5
6  Dim X As Long
7  Dim Y As Long
8
9  Dim T As Long
10
11 X = 100
12 Y = 300
13
14 'Long ... 4 bajty v paměti
15
16 CopyMemory T, ByVal VarPtr(X), 4
17 CopyMemory ByVal VarPtr(X), ByVal VarPtr(Y), 4
18 CopyMemory ByVal VarPtr(Y), T, 4
19
20 End Sub

```

Pozn. Funkci CopyMemory můžeme nasadit i na prosté kopírování z jedné proměnné do druhé, dokonce i na pole.

```
1  Sub KopieTextAPI()  
2  
3  'datový typ String  
4  
5  Dim X As String  
6  Dim Y As String  
7  
8  X = "Spejbl"  
9  Y = Space(Len(X))  
10  
11  CopyMemory ByVal StrPtr(Y), ByVal StrPtr(X), LenB(X)  
12  
13 End Sub
```

```
1  Sub KopieLongAPI()  
2  
3  'datový typ Long (4 bajty)  
4  
5  Dim X As Long  
6  Dim Y As Long  
7  
8  X = 123456  
9  
10 'CopyMemory Y, X, 4  
11  
12 'alternativně  
13 'CopyMemory ByVal VarPtr(Y), X, 4  
14  
15 End Sub
```

```

1  Sub KopiePoleLongAPI()
2
3  'pole, datový typ Long
4
5  Dim aPoleZdroj() As Variant
6  Dim aPoleCil() As Variant
7
8  aPoleZdroj = Array(1492, 3.1415, 65536, #8/19/2014#)
9  aPoleCil = Array(1, 2, 3, 4, 5)
10
11 'Variant ... 16 bajtů paměti pro číslo
12 'celkem 16 bajtů * počet prvků v poli
13 'tj. 16 * (UBound(aPoleZdroj) + 1)
14 'lépe
15 '16 * (UBound(aPoleZdroj) - LBound(aPoleZdroj)+ 1)
16
17 Dim X As Long
18 X = LenB(aPoleZdroj(0))
19
20 CopyMemory ByVal VarPtr(aPoleCil(0)), ByVal VarPtr(aPoleZdroj(0)), 16 * _
21     (UBound(aPoleCil) - LBound(aPoleCil) + 1)
22
23 End Sub

```

Pozn. Funkci CopyMemory lze uplatnit i v případě spojování (nabalování) řetězců.

```

1  Sub TechnikaNabalovaniAPI()
2
3  Dim aPoleRetezce() As Variant
4  Dim sRetezec As String
5  Dim i As Integer
6  Dim IPos As Long
7
8  aPoleRetezce = Array("M", "á", "n", "i", "č", "k", "a")
9
10 'vymezení délky budoucího řetězce
11 sRetezec = Space(UBound(aPoleRetezce) - LBound(aPoleRetezce) + 1)
12
13 'pro každý znak
14 For i = 0 To UBound(aPoleRetezce)
15
16     'nabalení znaku na řetězec
17     CopyMemory ByVal StrPtr(sRetezec) + IPos, ByVal StrPtr(aPoleRetezce(i)), _
18         LenB(aPoleRetezce(i))
19
20     'úprava pozice
21     IPos = IPos + LenB(aPoleRetezce(i))
22
23 Next i
24
25 End Sub

```

Do hry se také často zapojí úzké spojení řetězec-bytové pole. Ale to už je jiná kapitola.

Na závěr dnešního článku ještě dva tipy.

Dvoustavová proměnná

Dvoustavovou proměnnou je myšlena proměnná typu Boolean, v níž chceme cyklicky zaměňovat True na False a naopak. První z výpisů bych ohodnotil jedním bodem, druhý dvěma...

```
1      Sub DvoustavovaPromennaA()  
2  
3      'negace pravdivostní hodnoty  
4  
5      Dim bPromenna As Boolean  
6  
7      'bPromenna = ...  
8  
9      'obsahuje proměnná False?  
10     If bPromenna = False Then  
11         bPromenna = True  
12     Else  
13         bPromenna = False  
14     End If  
15  
16     End Sub
```

Stačí si uvědomit, že ve skutečnosti provádíme negaci pravdivostní hodnoty, pro kterou slouží ve VBA operátor Not. Ten vhodně doplníme deklarací proměnné jako Static.

```
1      Sub DvoustavovaPromennaB()  
2  
3      'negace pravdivostní hodnoty  
4  
5      'procedura si bude pamatovat hodnotu proměnné  
6      'při dalším volání  
7      Static bPromenna As Boolean  
8  
9      'negace operátorem Not  
10     bPromenna = Not bPromenna  
11  
12     End Sub
```

Dvoučíselná proměnná

Tímto pojmem míním proměnnou, která má nabývat jedné ze dvou číselných hodnot. Představte si formulář, který po klepnutí na tlačítko má odkrýt další nástroje, tj. rozbalit se. Fakticky upravujeme pouze jeho výšku (neberu nyní v potaz nějaký efekt rozbalení).

1	<b>Private Sub</b> CommandButton1_Click()
2	
3	'výchozí výška formuláře H = 100
4	'hodnoty H1 = 100, H2 = 300
5	'H1 + H2 = 400
6	
7	'změna výšky formuláře
8	Me.Height = 400 - Me.Height
9	
10	<b>End Sub</b>