INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

DUM 07 téma: Proměnné, konstanty a pohyb po buňkách ve VBA

ze sady:	**03**	**tematický okruh sady:**	**Tvorba skript a maker**
ze šablony:	10 Algoritmizace a programování	**určeno pro:**	4. ročník
vzdělávací obor:	18-20-M/01 Informační technologie - Aplikace osobních počítačů		
vzdělávací oblast:	odborné vzdělávání		
číslo projektu:	CZ.1.07/1.5.00/34.0066		
anotace:	DUM seznamuje s příkazy, které zabezpečují pohyb jednotlivých buňkách listu. Rovněž je zde vysvětleno načítání hodnot z a do buněk. Je zde vysvětleno vytváření proměnných či konstant a jejich datové typy.		
metodika:	viz metodický list VY_32_INOVACE_10307ml.pdf		
datum tvorby:	17.3.2013		

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek.

Materiál je publikován pod licencí Creative Commons

Proměnné v VBA

Jako každém programovacím jazyce se proměnná skládá: ze svého jména, datového typu a přiřazené hodnoty. Proměnná je tedy nějaké místo v operační paměti pro uložení hodnoty, které se může měnit.

Jméno nebo-li identifikátor

- Musí začínat písmenem.
- K dispozici je 256 znaků.
- Měl by být volen tak, aby bylo na první pohled jasné k čemu proměnná slouží.
- Nelze použít mezery a zakázané znaky či názvy z Windows.
- Nelze použít klíčová slova z Visual Basicu.

Datový typ

Jedná se o atribut proměnné, který určuje druh dat, co lze do ní vložit. Nemusí být vůbec definován a pak je proměnné přiřazen prostřednictvím VBA datový typ Variant, který může obsahovat jakákoliv typ dat. Datový typ Variant využíváme minimálně, protože výrazně zvětšuje nároky na paměť počítače a hodně zpomaluje běh počítače.

Konkrétní datové typy:

String – řetězec proměnné délky, který v paměti počítače zabere 10 byte + délku řetězce.

String * 32 – řetězec pevné délky, neobsazená místa budou doplněna nulami.

Byte – celočíselný datový typ, který může nabývat hodnot od 0 do 255. V paměti počítače zabere 1 byte, což je logické už z jeho názvu :-).

Integer – celočíselný datový typ, který má rozsah od -32 768 do +32 767. Na paměť počítače si nárokuje 2 byte.

Long – celočíselný datový ty, který má rozsah - 2 147 483 648 až 2 147 483 647. V paměti počítače zabírá už 4 byte.

Single – datový typ pro reálné číslo (desetinné číslo) s jednoduchou desetinnou přesností, který zabírá v paměti počítače 4 byte.

Double – datový typ pro reálné číslo s dvojitou přesností, který zabírá v paměti počítače 8 byte.

Currency – datový typ pro reálné číslo se čtyřmi desetinnými, které je ve formátu měny.

Date – jedná se o datový typ pro datum a čas, který zabírá 8 byte v paměti. Je vyjádřen pomocí reálného čísla, kde rozsah je 1. leden 0100 až 31. prosinec 9999.

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek. Materiál je publikován pod licencí Creative Commons

Boolean – logický typ, kde 0 znamená nepravda (False) a -1 či jakákoliv jiná hodnota než 0 znamená pravda (True). V operační paměti zabírá 2 byte.

Variant – jedná se o všeobecný datový typ. Použijeme ho pouze, když nelze stanovit, jakého datového typu proměnná bude. V paměti bude číslo zabírat 16 byte a řetězec 22 byte + délku řetězce.

Uživatelské datové typy – jsou definovány uživatelem. Využívají se při práci se skupinou datových položek, které k sobě nějakým způsobem patří. Jednotlivé položky mohou být rozdílných datových typů (např název – string, počet kusů – integer). Definujeme je pouze v deklarační části modulu. Pro definici užíváme klíčových slov Type a End Type

Definice uživatelské datové struktury

```
Type MojeStruktura

     Nazev As String * 32

     PocetKs As Integer

     Cena1Ks As Currency

End Type
```

Platnost proměnných

Před identifikátor proměnné zapisujeme jeden z následujících atributů: Static, Public, Private, Dim. Tyto atributy ovlivňují setrvání proměnné v paměti počítače a její viditelnost ostatními částmi kódu.

Static - Statická proměnná, hodnota je uchovávána i po skončení běhu procedury, do které proměnná patří.

Public – Je viditelná ve všech modulech a procedurách. Zachovává si platnost i po skončení dané procedury. Proměnou s tímto atributem zapisujeme do deklarační části modulu.

Private – Proměnná je dostupná pro všechny procedury v daném modulu. Proměnou s tímto atributem zapisujeme do deklarační části modulu.

Dim – Proměnná je dostupná pro jednu proceduru v daném modulu. Jen když tento modul běží. Nebo pro procedury v danném modulu, záleží na místě uvedení deklarace proměnné.

Definice proměnné¨

Atribut Identifikátor As Datový typ = Hodnota

```
Dim Vysledek As Single = 25,13
```

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons

```
Static Retezec As String = "Ahoj"
```

Pozor: Hodnotu řetězce píšeme do úvozovek!

Konstanty

Kormě proměnných existují v VBA také konstanty. Jsou to hodnoty uložené v paměti počítače, které mají svůj identifikátor. Jejich hodnota je pevně dána a nemůže se změnit. Deklarace probíhá stejně jako deklarace proměnné, pouze jako atribut před identifikátor zapíšeme klíčové slovo Const. Z hlediska ovlivňování jejich platnosti platí ta stejná pravidla jako v případě proměnných.

```
Const maximum As Long = 459

Public Const helpString As String = "HELP"

Private Const startValue As Integer = 5
```

Pohyb, čtení a zápis hodnot do buněk v Excelovském sešitě.

Abychom, se mohli po buňkách v sešitě pohybovat, je nejprve nutné nějakou buňku si nastavit jako aktivní. Existuje více možných cest, my si ukážeme cestu pomocí objektu range. K objektu range použijeme metodu select, která nám zabezpečí aktivaci buňky,

Konkrétní způsoby označení

`Range("A1").Select` - označení jedné buňky

`Range("1:1").Select` - označení celého řádku

`Range("A:A").Select` - označení celého sloupce

`Range("A1:D5").Select` - Označení souvislé oblasti buněk

Označení nesouvislé oblasti buněk

Range("A1", "D5").Select

Range("A1:B2", "C4:D5").Select

Označení dvou oblastí buněk

Range("A1:B2 , C4:D5").Select

Na první pohled je kód jednoduchý, ale v případě zapomenutí uvozovek kód není syntakticky špatně, ale skript nám pracuje špatně a tato chyba se velice obtížně hledá.

Výběr buněk mimo aktivní list.

Pro změnu listu použijeme kolekci sheets a metodu select.

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons

```
Sheets("List3").Select
```

Lze se též přesouvat mezi celými sešity a provádět spoustu dalších operací, toto je však nad rámec učiva látky střední školy.

Práce s aktivní buňkou

Používáme k tomu objekt ActiveCell a metodu FormulaR1C1. R1C1 znamená, v jakém formátu bude zadávána adresa tedy řádky a sloupce, proto si nejprve označíme buňku pomocí již zmíněného Range("R1C1").Select.

Čtení hodnoty z aktivní buňky

```
Dim cislo As Integer

cislo = ActiveCell.FormulaR1C1
```

Provádí se pomocí operátoru přiřazení. Jednoduše přiřadíme hodnotu v aktivní buňce do proměnné.

Zápis hodnoty do aktivní buňky

```
Dim hodnotaB As Integer

hodnotaB = 5

AciveCell.FormulaR1C1 = hodnotaB
```

Provádí se pomocí operátoru přiřazení. Jednoduše přiřadíme hodnotu z proměnné do aktivní buňky.

Pohyb po jednotlivých buňkách v sešitě

Použijeme k tomu následující metodu:

```
ActiveCell.Offset(x, y).Select
```

V závorce jsou znaky x a y místo niž zadáme čísla, která značí o kolik řádků (x) a o kolik sloupců (y) se má nastavit nová aktivní buňka.

`ActiveCell.Offset(1, 0).Select` – nová aktivní buňka je o jeden řádek níž

`ActiveCell.Offset(0, 1).Select` - nová aktivní buňka je o jeden sloupec doprava

`ActiveCell.Offset(-1, 0).Select` – nová aktivní buňka je o jeden řádek víš

`ActiveCell.Offset(0, -1).Select` – nová aktivní buňka je o jeden sloupec doleva

`ActiveCell.Offset(1, 1).Select` - nová aktivní buňka je o jeden řádek níž a o jeden sloupec doprava

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons

Zdroje:

Archiv autora

Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons