INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

DUM 05 téma: Úvod do VBA for application

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 seznamující se základní terminologií programovacího jazyka VBA a objektovým programováním obecně.		
metodika:	viz metodický list VY_32_INOVACE_10305ml.pdf		
datum tvorby:	11.3.2013		

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

Materiál je publikován pod licencí Creative Commons

Microsoft Visual Basic for Applications

Jedná se o moderní programovací jazyk obsažený v kancelářském balíku Microsoft Office, který je vhodný jak pro vytváření malých skriptů (maker na formátování textu), tak velkých aplikací (ručně naprogramovaného formuláře v databázi MS Access). Programovací jazyk VBA splňuje požadavky na moderní programování, konkrétně se jedná o objektivně orientovaný programovací jazyk a splňuje zásady strukturovaného programování.
Objektově orientované programovací jazyky respektive kód v nich napsaný je složen s tzv. objektů. Programování v nich spočívá v modelování nějaké části reality. Jednotlivé prvky modelované reality tvoří objekty kódu. Objekty kódu se seskupují do tzv. kolekcí. Objekty si pamatují svůj stav a navenek poskytují operace. Každý objekt pracuje jako černá skříňka, která dokáže provádět určené činnosti a komunikovat s okolím, aniž by vyžadovala znalost způsobu, kterým vnitřně pracuje. Objekt nemůže přímo přistupovat k „vnitřnostem“ jiných objektů. Jednotlivé objekty spolu komunikují pomocí rozhraní.
Strukturované programování lze definovat jako dodržování určitých zásad při tvorbě kódu. Tyto zásady poprvé přinesl programovací jazyk Pascal, který se objevil počátkem 70. let dvacátého století. Dnes tyto zásady při tvorbě kódu dodržujeme zcela automaticky aniž bychom si to uvědomovali, ale protože opakování je matka moudrosti tak si je zopakujeme.

Zásady strukturovaného programování:

Postup od shora dolů: Program (problém) je v podstatě pyramida, v jejímž vrcholu se nachází zadání. Řešení problému se směrem dolů postupně konkretizuje a narůstá do šířky. Výsledkem je pak konkrétní postup řešení daného problému. Tento způsob umožňuje snadnou lokalizaci chyby a její odstranění
Modularizace: Program je rozčleněn do jednotlivých samostatných částí bloků (modulů). Tyto moduly mají maximálně několik desítek řádků což nám umožnuje rozdělit program mezi programátory.
Stavebnicovost: Při tvorbě modulů musíme být co nejdříve univerzální, aby šel modul použít ve více projektech. Například modul pro komunikaci s tiskárnou napíšeme pouze jednou a co nejvíce univerzálně. Tento modul pak můžeme použít ve více programech.
Jelikož MS Excel má spoustu využití a mnohdy bychom potřebovali vytvořit nějaké složitější makro, které nemůžeme nahrát pomocí záznamníku maker, tak se budeme zabývat tvorbě maker v prostředí MS Excelu.

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

Základní pojmy

Objekt

V prostředí Excelu je myšleno pojmem objekt např. sešit, list, buňka, graf ... U všech objektů existují metody a vlastnosti. V kódu VBA zapisujeme název objektu a napravo od něj zapisujeme vždy metody a vlastnosti patřící k objektu. Na objekt odkazujeme v rámci kolekce (indexem nebo jménem) nebo objektu předchází slovo Active a bezprostředně za ním název objektu.

Např.: `Workbook, ActiveWorkbook`

Metoda

Zobrazuje akci, kterou lze s objektem provádět, např.: otevřít, zavřít, přidat, odstranit. Metodě musí předcházet objekt, se kterým se má akce provést. Prakticky ji označuje sloveso.

Např.: `Worksheets("List2").Delete`

- odstraní list List2

Vlastnost nebo-li Atribut

Nabývá nějakých hodnot např. jméno, velikost,… Vlastnosti popisují objekty v prostředí VBA. Vlastnosti od objektů lze zjistit nebo nastavit.

Např.: `MsgBox Worksheets(2).Name`

- zobrazí název listu a příkazem

Např.: `Worksheets(2).Name = "Pokus"`

- list přejmenuje na Pokus.

Kolekce

Jedná se o množinu objektů patřící do stejné skupiny (třídy). U kolekcí stejně jako u objektů jsou vlastnosti a metody. Používají se pro práci s celými množinami nebo jen s jednotlivými prvky kolekce. Na prvky kolekce se lze odkazovat indexem tzn. pořadovým číslem v kolekci nebo pomocí jména prvku. Jednoduchá memotechnická pomůcka je, že končí na písmeno s.

Např.: V excelovském sešitě se 3.list jmenuje List3:

`Worksheets (3)`

– pomocí indexu

`Worksheets ("List3")`

– pomocí jména

- Tyto dva zápisy jsou ekvivalentní, jedná se o třetí list z kolekce listů.

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

Material je publikován pod licencí Creative Commons

Analogie

Pro přiblížení teď vyložených pojmů si teď ukážeme malou analogii. Knihovna je kolekce jednotlivých objektů knih. Na konkrétním místě se v ní nachází kniha XY. Místo, kde se kniha nachází má nějaký pořadový index např. 5005. Zápis Knihovna(5005) odpovídá knize XY. Knihy jak už bylo zmíněno tvoří jednotlivé objekty knihovny. U knihy jsou definovány její vlastnosti jméno, velikost, počet listů… Metoda pak lze přirovnat k otevření knihy.

Např.: `Knihovna(5005).Name`

- vrací název naší knihy.

Konkrétní objekty ve VBA

V programu musíme vždy určit objekt, se kterým budeme pracovat. Z hlediska MS Excelu je Sešit je objekt Workbook, obsahuje kolekci listů Worksheets. Font je objektem vůči Bold, Italic, Color, Size …

Např.: `Worksheets("List3").Range("A3:G3").Font.Bold = True`

- Oblast buněk se zformátuje tučným řezem písma.

Font je vlastností vůči oblasti, která je objektem, jak je vidět jde o směr pohledu. Objekt je vždy zapsán vlevo. Z tohoto důvodu je vhodné číst příkazy zprava doleva, to vede k lepšímu pochopení příkazů.

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