

APLICAREA MACRO-DEFINIȚIILOR LA EXPORTUL DE DATE DIN MICROSOFT EXCEL ȘI MICROSOFT POWER POINT PENTRU ANALIZA REUȘITEI STUDENȚILOR

Vitalie ȚÎCĂU, asistent univ.

Facultatea de Științe Reale, Economice și ale Mediului,
Universitatea de Stat „Alec Russo” din Bălți

Abstract: *This article research the possibilities of automated export of tables and diagrams from Microsoft Excel to Microsoft Power and programming macro-definition. It describes ways to programming in Visual Basic for Application, to applied VBA in Microsoft Excel and Microsoft Power Point. Also is describes how to program macro-definitions to analyze student success.*

Keywords: *macro-definitions, programming in Visual Basic for Application, student success analysis, spreadsheet processor Microsoft Excel, Microsoft Power Point.*

Introducere

Din momentul elaborării, Microsoft Excel a devenit un produs foarte popular în ceea ce privește calculul tabelar. Aplicația Microsoft Excel se modifică permanent de la o versiune la alta. Cu toate acestea, mulți utilizatori folosesc doar o mică parte din capacitățile acestui procesor. Programarea în Excel se realizează prin limbajul de programare Visual Basic for Application. VBA este un subset de limbaj de programare Visual Basic puternic și este inițial încorporat în majoritatea aplicațiilor Office. De asemenea, VBA este folosită și în Microsoft Power Point.

Comenzile Macro reprezintă programe scrise în Visual Basic for Application. Pentru a automatiza o activitate repetitivă, se poate înregistra o macrocomandă cu înregistratorul de macrocomenzi în Microsoft Excel sau Microsoft Power Point. Se poate înregistra o macrocomandă, care va aplica cu formatul dorit și apoi va reda repetat macrocomandă ori de câte ori este necesar. Datorită macrocomenzilor, procesorul de calcul tabelar Microsoft Excel se dezo-voltă singur, adaptându-se la cerințele unui anumit utilizator.

La gestiunea reușitei studenților sau a elevilor, pentru a analiza rezultatele evaluării, deseori apare necesitatea de export de date din Excel în Power Point pentru o interpretare grafică a acesteia.

În lucrare este cercetată următoarea problemă: exportul de date și diagrame, stocate în Microsoft Excel în prezentările Microsoft Power Point prin programarea unor macro-definiții.

Ca aplicare a materialului cercetat s-au elaborat automatizări la prezentarea și analiza reușitei studenților.

Programarea în VBA Excel. Obiecte, colecții, proprietăți și metode

Microsoft Excel este un program de calcul tabelar, ține evidența mai multor date (numere, de tip text, calendaristice etc.) și realizează diverse operații cu acestea. Poate fi utilizat pentru calcule ce implică manipularea unor volume relativ mari de date. Aplicațiile de calcul tabelar sunt programe specializate pentru a prelucra date organizate sub formă de tabele.

Programarea în Excel se realizează prin limbajul de programare Visual Basic for Application (VBA), care este inițial încorporată în cel mai faimos procesor de masă de la Microsoft [1]. După cum arată practica, chiar și utilizatorii care nu au abilitățile de programare profesională pot stăpâni noțiunile de bază ale VBA. Caracteristicile VBA includ execuția de scripturi într-un mediu de aplicații de birou.

Dezavantajul programului sunt probleme asociate compatibilității diferitor versiuni. Ele se datorează faptului că codul programului VBA se referă la funcționalitatea prezentă în noua versiune a produsului, dar nu și la vechea versiune. De asemenea, dezavantajele includ deschiderea excesiv de mare a codului care trebuie schimbat de un outsider. Cu toate acestea, Microsoft Office și IBM Lotus Symphony permit utilizatorului să utilizeze criptarea codului inițial și să stabilească o parolă pentru a o vizualiza. Editorul Visual Basic este un reprezentant de

încredere al VBA pe ecran - aici se scriu programele, se construiesc ferestrele care se deschid cu aceste programe și se testează codul pentru a verifica corectitudinea lui.

VBA este un limbaj de programare încorporat în multe programe, de la aplicațiile Microsoft Office, Microsoft Project, Visio și AutoCAD la numeroase aplicații specializate concepute pentru gestionarea proceselor de producție, contabilitate, pentru resurse financiare sau suport pentru informații despre clienți.

Aspectele esențiale ale programării în VBA:

- folosirea instrumentelor de programare vizuală VBA;
- înregistrarea și editarea macrocomenzilor;
- lansarea programelor VBA din alte aplicații;
- crearea de cutii de dialog și alte elemente de interfață;
- lucrări cu obiecte, care vor fi cheia utilizării întregii puteri a aplicațiilor VBA.

Toate versiunile VBA, de la versiunea 6 la versiunea 6.3, sunt complet identice din punctul de vedere al programatorului [2]. Microsoft a rezolvat câteva erori, performanțe îmbunătățite, însă modalitățile de scriere a codului sau de creare de casete de dialog nu s-au schimbat.

Pentru a utiliza VBA, trebuie să fie instalată cel puțin o aplicație, în care sunt încorporate instrumentele de dezvoltare ale programului VBA. Printre aceste aplicații: Microsoft Office, urmat de grupul tot mai mare de produse aparținând altor producători [2]: orice aplicație Microsoft Office – Word, Excel, PowerPoint, Access, Outlook sau FrontPage; Microsoft Project; CorelDraw versiunea 9 sau 10 și Corel WordPerfect Office 2000; AutoCAD R14, AutoCAD 2000 sau AutoCAD 2000i pentru Windows etc.

În Excel un *obiect* este foaia, registrul de lucru, celula și intervalul. Aceste obiecte au o ierarhie specială, adică se supun reciproc [1]. Principală este aplicația corespunzătoare programului Excel. Apoi urmează cărțile de lucru, foile de lucru și, de asemenea, gama. De exemplu, pentru a accesa celula A1 pe o anumită foaie, trebuie de specificat calea, luând în considerare ierarhia. În ceea ce privește conceptul de „colectare”, este vorba de un grup de obiecte din aceeași clasă, care în înregistrare are forma *ChartObjects*. Elementele sale individuale sunt, de asemenea, obiecte.

Următorul concept sunt *proprietățile*. Ele reprezintă o caracteristică necesară a oricărui obiect. De exemplu, pentru interval este o valoare sau o formulă. *Metodele* sunt comenzi care arată ce trebuie de făcut. Când se scrie codul în VBA, metodele trebuie separate de obiect printr-un punct. De exemplu, deseori se folosește comanda *Cells(1, 1).Select*. Aceasta înseamnă că trebuie să selectați o celulă cu coordonate. Împreună cu aceasta se folosește adesea *Selection.ClearContents*. Execuția sa înseamnă ștergerea conținutului celulei selectate.

Pentru a lansa aplicația VB se utilizează combinația de taste „Alt” și „F11”. În bara de meniuri din partea de sus a ferestrei se face clic pe pictograma de lângă pictograma Excel, apoi se alege comanda *Module* și se salvează, făcând clic pe pictograma cu imaginea necesară, după care se scrie o schiță a codului [2]. Se poate scrie orice cod și crea un nou instrument în VBA Excel. Utilizatorii familiarizați cu elementele de bază ale Visual Basic mult mai ușor vor aplica facilitățile VBA. Cu toate acestea, chiar și cei care nu le au, dacă doresc, vor putea să se stabilească suficient de repede.

Macro-uri VBA Excel

Macrourile Excel sunt programe, scrise în Visual Basic for Application. Astfel, programarea în Excel este crearea de macrocomenzi cu codul necesar. Având în vedere modul de creare a modulelor pentru scrierea de macrocomenzi, să cercetăm exemple specifice ale programelor Excel VBA.

Exemplul 1. De scris un program, care să copieze valoarea conținutului unei celule.

Pentru aceasta: se deschide fila *View* și se accesează pictograma *Macro*, se alege *Record new macro* și se completează formularul deschis. Pentru simplitate, *Macro1* este lăsat în câmpul *Nume macro* și se recomandă introducerea unei combinații de taste.

Deoarece înregistrarea macro este deja pornită, se copiază conținutul unei celule într-alta. Se revine la pictograma originală, se face clic pe *Stop recording*, această acțiune înseamnă

finalizarea programului. Pentru lansarea macroului se revine la opțiunea *Macro* și se selectează din listă *Macro 1*, se face clic pe *Run* sau se folosește combinația de taste [2].

Rezultatul este o acțiune care a fost efectuată în procesul de înregistrare a unei macrocomenzi.

Pentru a vedea codul, se revine la *Macro* și se face clic pe *Edit*. Ca urmare, ele ajung în mediul VBA. De fapt, macrocomanda în sine este localizată între liniile *Sub Macro1 ()* și *End Sub*. În cazul în care copierea a fost efectuată, de exemplu, de la celula *A1* la celula *C1*, atunci una dintre liniile de cod va arăta ca *Range(„C1”).Select* [3]

Buclele VBA ajută la crearea unor macrocomenzi variate în Excel [4], ce țin de repetarea unor acțiuni. De asemenea, în viața de zi cu zi, de cele mai multe ori este nevoie de luat o decizie sau alta, în funcție de anumite condiții. Instrucțiunile de decizie sunt și în VBA Excel. Exemple de programe în care se alege progresul ulterior al algoritmului, mai degrabă decât predeterminat inițial, cel mai adesea se utilizează structura *If ...* Construcția pentru cazurile complexe *If ... Then ... Else* [5].

Programarea în procesorul Microsoft Excel nu este atât de dificilă. Mai ales, dacă se utilizează funcțiile VBA. În total, acest limbaj de programare, creat special pentru scrierea de aplicații în Excel și Word, conține aproximativ 160 de funcții [1].

Crearea unui șablon. Toate acțiunile sunt efectuate pe o foaie standard în Excel. Celulele libere sunt rezervate pentru introducerea datelor cu privire la numele companiei de consum, suma costurilor, nivelul lor și cifra de afaceri. Întrucât numărul companiilor (societăților) pentru care se compune raportul nu este fix, celulele nu rezervă în avans pentru a introduce valorile rezultatelor și numele complet al specialistului. Fișei de lucru i se atribuie un nume nou. De exemplu, „*Raport*”.

În Excel VBA, un set de comenzi pentru a efectua o anumită sarcină este plasat într-o procedură, care poate fi o procedură funcțională sau o subprocedură. Principala diferență între o procedură funcțională VBA și o procedură *Sub* este că o procedură funcțională returnează un rezultat, în timp ce o procedură *Sub* nu o face [5]. Prin urmare, pentru a efectua o sarcină care returnează un rezultat, se poate folosi în general o procedură funcțională, dar, dacă este nevoie de un set de acțiuni care trebuie efectuate, atunci se alege o procedură *Sub*.

Procedurile VBA pot fi transmise prin intermediul *argumentelor*, care sunt declarate în definiția procedurii. Nu este esențial ca funcțiile VBA sau *Sub* să aibă argumente.

Atunci când argumentele sunt transmise procedurilor VBA, ele pot fi transmise prin valoare sau prin referință – se poate specifica utilizând cuvintele cheie *ByVal* sau *ByRef*. Implicit, argumentele VBA sunt transmise prin referință.

Valorile returnate au următoarele reguli [6]:

- tipul de date al valorii returnate trebuie declarat în antetul funcției;
- valoarea care trebuie returnată trebuie să fie atribuită unei variabile care are același nume ca și funcția. Această variabilă nu trebuie declarată, deoarece există deja ca parte a funcției;
- puteți apela funcțiile VBA dintr-o foaie de lucru Excel, la fel cum puteți apela oricare dintre funcțiile Excel integrate.

Aplicarea macrourilor la rezolvarea unor sarcini

Gruparea datelor pe specialități. Se cunosc rezultatele de la evaluare a studenților la sesiunea de vară / iarnă de la o facultatea a unei Universități. Datele sunt înscrise în următoarele coloane într-o foaie de calcul Excel (figura 1): grupa, nr. de studenți total, clasificarea numărului studenților în grupe: numărul de studenți cu note din categoriile A, B, C, D, E, F, numărul de studenți restanțieri, exmatriculați, precum și multe alte informații.

Sarcină: Să se selecteze în foi de calcul aparte informațiile necesare pentru prezentarea reușitei studenților și gruparea pe specialități, calculul % reușitei pentru fiecare grupă.

Nr. de stud. neadmiși la sesiune																							
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Rezultatele sesiunii de examinare pe anul de studii 2019-2020																							
Universitatea de Stat „Alecu Russo” din Bălți																					sesiunea de iarnă		
													Reușita						Restanțieri				
													A	B	C	D	E	FX	F				
													9,01-10,0	8,01-9,0	7,01-8,0	6,01-7,0	5,0-6,0	3,01-4,99	1,0-3,0				
																				Total	sesiunea prelungită	Exmatriculați	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
Facultatea Științe Reale, Economice și ale Mediului																							
Specialitatea Educație tehnologică ET																							
II	10	6	4			10	8	2		8					7	1							2
III	13	7	6			13	12	1		12					3	9					1		
IV	10	7	3			10	8	2		8				1	5	2					2		
Total	33	20	13			33	28	5		28				1	15	12				3		2	
Specialitatea Design vestimentar industrial																							
III	7		7			7	5	2		5				2	2	1					2		
IV	5		5			5	5			5				4		1							
V	1		1	1																			
Total pe speciali-tate:	13		13	1		12	10	2		10				6	2	2				2			
Specialitatea Inginerie și management (în transport auto) IM																							
I	32		32			32	25	7	5	4	16	3		1	3	3					16		9

Figura 1. Rezultatele de la evaluare a studenților

S-a creat un macrou *Specialitati*. Selectarea informației se bazează pe faptul că identificatorul specialității – abrevierea – este scris pe dreapta celulei, care începe cu cuvântul „Special”. Se parcurg toate rândurile foii de calcul „Total”, începând cu primul rând cu date – rândul 9. Dacă se întâlnește celula ce începe cu „Total”, atunci datele la specialitatea cercetată s-au epuizat. Apoi se creează o nouă foaie de calcul și se transferă datele noii specialități pe această foaie.

Sub *Specialitati()* ' *Specialitati Macro*, ' *Keyboard Shortcut: Ctrl+q*

Sheets("Total").Select

For i = 9 To WorksheetFunction.CountA(Columns(2))

If Left(Cells(i, 2), 7) = "Special" Then

k1 = i

Cells(i, 1) = Right(Cells(i, 2), 2)

grupa = Cells(i, 1)

ElseIf Left(Cells(i, 2), 5) = "Total" Then

ThisWorkbook.Sheets.Add.Name = Cells(k1, 1)

ActiveCell.FormulaR1C1 = Sheets("Total").Cells(k1, 2)

Cells(1, 1) = Left(Sheets("Total").Cells(k1, 2), n - 3)

Cells(2, 1) = "Grupa"

Cells(2, 2) = "A"

Cells(2, 3) = "B"

Cells(2, 4) = "C"

Cells(2, 5) = "D"

Cells(2, 6) = "E"

Cells(2, 7) = "R"

Cells(2, 8) = "Exm."

Cells(2, 9) = "Mob/C.a."

Cells(2, 10) = "Media"

Cells(2, 11) = "Nr"

sir1 = "Reusita: "

For j = k1 + 1 To i - 1

m = j - k1 + 2

Cells(m, 1) = Sheets("Total").Cells(j, 1)

Cells(m, 11) = Sheets("Total").Cells(j, 3)

Cells(m, 2) = Sheets("Total").Cells(j, 15)

Cells(m, 3) = Sheets("Total").Cells(j, 16)

...

Next j

Cells(m + 1, 1) = sir1

Sheets("Total").Select

Cells(i, 1) = "Ok"

Else

Select Case Cells(i, 2)

Case "I": sir = "1"

Case "II": sir = "2" ...

End Select

Cells(i, 1) = grupa + sir + "1R"

End If

Next

End Sub

Gruparea rezultatelor pe ani de studii. Să se selecteze în foi de calcul aparținând informațiile necesare pentru prezentarea reușitei studenților și gruparea pe ani de studii.

S-a creat un macro *AndeStudii*. Selectarea informației se bazează pe faptul că în coloana 2 este indicat anul de studii cu cifre romane. Conform literei din celula A8 (R / Z / M) se generează datele pentru studiile de licență cu frecvența redusă sau la zi, sau pentru studiile de master.

Trasarea automatizată a diagramelor pentru fiecare specialitate în parte

Una dintre principalele utilizări ale Excel constă în analiza datelor. Pentru analiza datelor este adesea convenabil să folosiți diagrame cu caracteristicile lor speciale, cum ar fi tendințele. În practică, apar foarte des sarcinile de automatizare a creării multor diagrame similare (de obicei, bazate pe informații obținute din baza de date). Graficele în Excel sunt create folosind obiectul *Chart*.

S-a creat un macro *Diagrame*, în care din foile de calcul (cu excepția celor ce încep cu „Total” sau „An”) se selectează regiunea A2:Ik, unde k – numărul ultimei înregistrări cu date, calculat:

$k = \text{Sheets}(\text{nume}).\text{Columns}(1).\text{End}(\text{xlDown}).\text{Row} - 3$

Numele diagramei este stabilit similar cu numele foii de calcul, folosind proprietatea *Name*. Rezultatul apelului macroului este prezentat în figura 2.

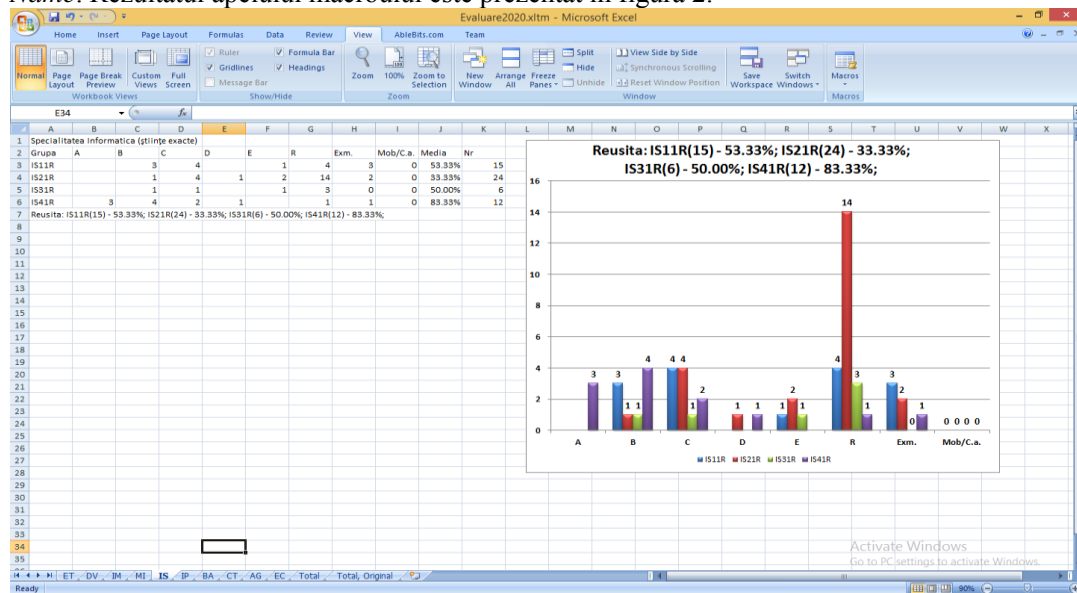


Figura 2. Generarea diagramelor în foile de calcul

Exportul automatizat de tabele și diagrame din Microsoft Excel în Microsoft PowerPoint

Sarcina transferului de grafice, diagrame, tabele din Excel într-o prezentare PowerPoint este complicată din faptul că în cele mai recente versiuni ale editorului de prezentare (2007, 2010, 2015, ...), dezvoltatorii au eliminat capacitatea de a înregistra acțiuni efectuate într-un macro.

S-a creat un macro complex *ExportExcelPowerPoint*, care realizează mai multe acțiuni, grupate în câteva etape.

1. Completarea primului slide. Se folosește crearea obiectului pentru o prezentare Power Point, aplicând metodele *CreateObject* și *Add*. Apoi se creează un slide nou aplicând metoda *CreateObject Slides.Add(1, 1)*, stilul fiind 1. De asemenea, se completează cu caracteristicile necesare textului *PPSlide.Shapes(1)* și *PPSlide.Shapes(2)* pentru titluri.

2. Completarea următoarelor slide-uri cu tabele referitor la anii de studii. În funcție de numărul de foi de calcul cu ani de studii (5/4/2) se creează slide-uri noi aplicând metoda *CreateObject Slides.Add(nr, 2)*, stilul fiind 2 (cu tabele).

3. Completarea următoarelor slide-uri cu diagrame referitor la specialități. În funcție de numărul de foi de calcul cu specialități, generate la pasul 1 prin macroul „*Specialitati*” se creează slide-uri noi aplicând metoda *CreateObject Slides.Add(nr, 8)*, stilul fiind 2 (cu diagrame).

4. Salvarea prezentării.

Concluzii

Foile de calcul Microsoft Excel se folosesc pe larg pentru stocarea diverselor informații.

Utilizarea macro-definițiilor facilitează gestiunea dalelor din foile de calcul Microsoft Excel și Microsoft Power Point și optimizează timpul de prelucrare a datelor.

Deși VBA oferă capacitatea de a automatiza procesele din interiorul aplicațiilor Office și între acestea, de multe ori nu este necesar să cunoașteți cod VBA sau programare, deoarece înregistratorul de macrocomenzi face diverse operații necesare în mod automat.

Exemplele de macro-definiții elaborate pot fi utile și la rezolvarea unui set de probleme similare în alte domenii.

Bibliografie:

1. MANSFIELD, R. *Office 2003 Application Development All-in-One Desk Reference For Dummies*. Wiley Publishing, Inc., Indianapolis, Indiana, 2004. 834 p. ISBN: 0-7645-7067-6
2. SHEPHERD, R. *Excel VBA Macro Programming*. McGraw-Hill/Osborne, California U.S.A., 2004. 340 p. ISBN: 0072231440
3. *Macro-uri în Excel: introducere*. [online] [citat 10.09. 2020]. Disponibil: <https://excelninja.ro/macro-uri-in-excel-introducere/>
4. *VBA Excel: exemple de programe. Macrografice în Excel*. [online] [citat 14.09. 2020]. Disponibil: <https://olnafu.ru/calculatoare/17335-vba-excel-exemple-de-programe.html>
5. *Automatizarea activităților cu înregistratorul de macrocomenzi*. [online] [citat 30.10.2020]. Disponibil: <https://support.microsoft.com/ro-ro/office/automatizarea-activit%C4%83%C8%9Bilor-cu-%C3%AEnregistratorul-de-macrocomenzi-974ef220-f716-4e01-b015-3ea70e64937b>
6. *Crearea, aplicarea și editarea macrocomenzilor*. [online] [citat 05.10.2020]. Disponibil: <http://www.scribub.com/stiinta/informatica/word/crearea-aplicarea-si-editarea-63979.php>