

Wprowadzenie do edycji makr w Excelu

Zakładam, że masz otwarty edytor VBA. Jeśli nie, to wciśnij proszę w Excelu ALT+ f11 – ten skrót nie zmienił się na szczęście w Excelu 2007.

Makra można wpisywać w kilku miejscach (o tym kiedy indziej). Nagrywanie makra powoduje automatyczne dodanie nowego modułu. My dodamy go ręcznie poprzez wybór w menu edytora „INSERT->MODULE „
Nagrałem przykładowe makro, żebyśmy mieli co omawiać. Oto jego wynik, który możesz wkleić do edytora :

```
Sub Makro1()  
,  
' Makro1 Makro  
,  
,  
  
ActiveCell.value = "ala ma kota"  
Range("A1").Select  
Selection.Copy  
Range("H12").Select  
ActiveSheet.Paste  
Rows("11:11").Select  
Selection.EntireRow.Hidden = True  
Columns("B:B").Select  
Selection.FormulaR1C1 = "1"  
Sheets.Add After:=Sheets(Sheets.Count)  
End Sub
```

Po wciśnięciu klawisza f8 kolejne kroki są wykonywane pojedynczo. Excel dopuszcza edytowanie makra w pewnym zakresie pomiędzy wciśnięciami f8. Możesz także ustawiać znacznik wykonywania w dowolnym punkcie programu, pomijać jego części, lub powracać do wcześniejszych fragmentów. Jest więcej opcji, ale nie zdążę wszystkich przedstawić.

Takie makro możemy w prawdzie wykorzystać, ale nie przyda nam się na wiele. Ktoś stworzy ważny plik, zobaczy, że jest makro o nic nie mówiącym tytule, odpali je i zniszczy ważne dane (dobrze, że makro nie zapisuje pliku). Zaczniemy więc od nazwy.

Sub - określa nam jedną procedurę – makro, czyli listę poleceń do wykonania. Taki blok kończy się słowami „end sub „. Nazwy w VBA mogą składać się z liter, cyfr i podkreślników, jednak zaczynać mogą się tylko od liter. To, co jest w nawiasach to parametry- na chwilę to zostawimy. Powiem tylko, że jeśli są jakieś parametry, to makro nie będzie widoczne w liście makr (ale nie tylko wtedy). Zmień nazwę makra

W kolejnych kilku liniijkach są komentarze. Wszystko w linijce po znaku ‘ jest komentarzem. W tym wypadku warto by było wpisać po nazwie makra krótki jego opis. VBA nie posiada

czegoś takiego jak komentarze blokowe. Oznacza to ,że jeśli chcesz pisać komentarz w kilku liniach na początku każdej będziesz musiał dodać znak apostrofu.

Zanim przejdziemy dalej, chciałbym zwrócić Twoją uwagę na drugie użycie komentarzy. W mojej pracy bywa tak, że muszę na chwilę wyłączyć część działającego kodu. Ponieważ nie mam do dyspozycji komentarzy blokowych, muszę wstawić na początku każdej linii, którą chcę chwilowo wyłączyć apostrof. Mógłbym robić to ręcznie, ale edytor może zrobić to za nas. Klikamy więc VIEV-> TOOLBARS->EDIT. Pasek ,który się pojawił powinien być, moim zdaniem, domyślnie włączony. Na początku mojej pracy z Excelem wstawiałem apostrofy ręcznie, co było uciążliwe. Na pasku masz przycisk Comment Block, z pomocą którego możesz zakomentować zaznaczony fragment kodu.

Przyjrzymy się pierwszej linijce z kodem. Jest to :

```
ActiveCell.value = "ala ma kota"
```

Oznacza to tyle co ustaw wartość w aktywnej komórce na słowa „Ala ma kota” Jeśli znasz język angielski, czytanie kodu prostych makr powinno być dla ciebie dość łatwe.

Activecell to aktywna komórka. Jeśli odpalisz makro, zdanie „Ala ma kota” pojawi się zawsze w komórce , którą miałeś zaznaczoną. Możesz to przetestować. Do składowych obiektu, czy to składnika (czyli zmiennej) , czy metody (czyli funkcji składowej) odwołujemy się za pomocą kropki. Jeśli wydało ci się to skomplikowane, to pomyśl po prostu, że jeśli chcesz coś zrobić z obiektem to używasz do tego kropki. W kolejnych wierszach także widzimy użycie kropki, czyli robimy coś na obiektach:

```
Range("A1").Select ' zakres „A1” zaznacz  
Selection.Copy ' Zaznaczenie skopiuj  
Range("H12").Select ' zakres „h12” zaznacz  
ActiveSheet.Paste' W aktywnym arkuszu wklej ( po prostu wklej tu gdzie jesteś )  
Rows("11:11").Select 'zaznacz rzędy od 11 do 11  
Selection.EntireRow.Hidden = True ' W zaznaczeniu ,wybierz wszystkie rzędy i ustaw  
wartość ukryte na prawda.  
'.....
```

I tak dalej .

Nawet jeśli nie rozumiesz wszystkiego , nie przejmuj się . Gdyby nauka przychodziła nam bez trudu , to pewnie wszyscy kochalibyśmy szkołę i byli geniuszami. Korzystaj z dostępnej w edytorze pomocy, jest dobrze napisana, choć na początku może sprawiać spore problemy. Po pewnym czasie powinienes z łatwością odnajdywać to, czego szukasz i będziesz rozumiał skomplikowane na pierwszy rzut oka podpowiedzi.

Wróćmy jednak do naszego makra. Nagrywając je chciałem ułatwić sobie wpisywanie tekstu w pierwszych dziesięciu polach (od A1 do A10). Niestety, ponieważ jestem nie doświadczony, po drodze wykonałem sporo różnych zbędnych operacji, a makro nie robi tego, co chciałem. Niestety często tak bywa. Jesteśmy zmuszeni do edytowania makra, i do wyciągnięcia z tego co się nagrało, tylko tego co jest nam potrzebne.

Analizując jeszcze raz kod, doszedłem do wniosku, że większość linijek jest zbędna i robią tylko bałagan. Skasować musiałem wszystko, oprócz pierwszej linijki, a i ta nie działa prawidłowo. Nie oznacza to, że niczego się nie nauczyłem. Poznałem podstawowe obiekty, takie jak:

Range – Zakres

Sheet – Lista Arkuszy

ActiveSheet – Aktywny Arkusz

Selection – zaznaczenie

Columns – kolumny

Ta wiedza przyda nam się w kolejnych makrach. Nagrywanie makr to dobry sposób nauki. Widzimy co krok po kroku zrobił program. Trzeba jednak być bardzo ostrożnym, żeby nie nagrywać niepotrzebnych rzeczy i nie zaciemniać sobie kodu.

Chciałem mieć w polach a1 do a10 napis „Ala ma kota”. Została mi tylko linijka :

```
ActiveCell.value = "ala ma kota"
```

No i co? Będę zaznaczał kolejne komórki i uruchamiał makro? To ja dziękuję za taką pomoc! Przypadek, który omawiamy jest trywialny i szybciej wykonalibyśmy go „na piechotę” (np. zaznaczając komórki, wpisując „Ala ma kota” i wciskając ctrl+enter), ale na razie się uczymy. Najkrótszy kod, którym moglibyśmy rozwiązać ten problem to :

```
Range(„a1:a10”).value = „Ala ma kota”
```

Przetestuj. Czy to działa? Jest sukces! To co? Kończymy? O nie! Pomęcę was jeszcze. Nagle zachciało mi się, żeby w kolejnych linijkach wpisana była liczba przed zdaniem, i żeby te liczby były 2 razy większe niż numer wiersza.

Będziemy musieli zmodyfikować nasze makro. Masz już podstawową umiejętność edycji makr. Wiesz, że z obiektami możesz robić różne rzeczy. Im więcej będziesz poznawał obiektów, tym więcej rzeczy będziesz w stanie wykonać.

Czego nam brakuje? Zmiennych, pętli i instrukcji warunkowych.

W tak krótkim tekście nie poznamy wszystkiego, ale chcąc zarazić cię tematem pokażę wykorzystanie jednej pętli i jednej tylko zmiennej. Orientujesz się pewnie, co to są zmienne. Wiesz coś o typach zmiennych, albo o deklaracji zmiennych? Jeśli nie, to na razie nie szkodzi. Choć dobrą praktyką jest deklarowanie zmiennych (moim zdaniem), to w VBA nie musisz tego robić. Nie musisz też definiować ich typu (choć to pomaga przy obiektach). Co to oznacza? Że jeśli chcesz użyć zmiennej to po prostu jej używasz. Przykład:

```
X = 2
```

Takiej zmiennej możesz użyć jako tekstu, lub jako liczby (ma typ variant)
Modyfikować zmienne możesz w ten sam sposób

X= X+1

Czyli jeśli x= 1 to po wykonaniu tej linijki x będzie wynosić 2

Żeby rozwiązać nasz problem będę musiał zrobić coś takiego:

Licząc od jednego do dziesięciu wpisuj w kolejne wiersze numer liczby pomnożony przez 2 i „Ala ma kota”

Wykorzystamy pętlę for

For to taka pętla, którą przetłumaczyłbym jako:” Dla zmiennych, od pewnej wartości do kolejnej, wykonaj zadane polecenia.”

To niefortunne tłumaczenie, dla mnie jednak prościej wygląda w kodzie. Na przykład :

```
For x = 1 to 10  
' Tu będą instrukcje  
Next x
```

Taka pętla zaczyna się od słowa for , później "mówimy" komputerowi od ilu do ilu ma liczyć, a następnie pokazujemy mu , gdzie kończą się instrukcje które ma wykonać (next x)

Czyli komputer policzy od 1 do 10 i wykona tyle razy to co jest w środku.

Tego potrzebujemy! Pozostaje nam wydrukować wartości. Po wstawieniu naszego kodu drukującego mamy takie coś:

```
For x = 1 to 10  
ActiveCell.value = "ala ma kota"  
Next x
```

Jak dodać do tego numer wiersza razy 2? Wszystkie teksty w VBA łączymy za pomocą znaku & (pamiętamy o dodaniu spacji) . Czyli nasz kod mógłby wyglądać tak:

```
ActiveCell.value = x & " ala ma kota" wydrukuj x i spację i „Ala ma kota”
```

Wystarczy, że pomnożymy x*2 i będziemy mieli to, o co nam chodziło:

```
ActiveCell.value = x *2 & " ala ma kota" wydrukuj x*2 i spację i „Ala ma kota”
```

[to jest nie eleganckie rozwiązanie]

Jeśli puścisz makro instrukcja po instrukcji (F8), zauważysz, że makro drukuje 10 razy tekst w pierwszej linijce. Brakuje nam przejścia do kolejnej linijki. Przy pierwszym podejściu zrealizujemy to za pomocą metody offset.

Offset to taka funkcja , która wskazuje na komórkę przesuniętą , o pewną liczbę komórek w danym kierunku. Przykład zilustruje to lepiej.

```
selection.offset(0,1).select
```

zaznaczy komórkę w prawo o jeden od zaznaczonej. Pierwsza liczba przesuwana w dół, druga w lewo. (w górę i w prawo można przesuwać wpisując liczby ujemne)
My chcemy komórkę w dół więc nasz kod będzie wyglądał tak:

```
For x = 1 to 10  
ActiveCell.value = x *2 & " ala ma kota"  
selection.offset(1,0).select  
Next x
```

Działa! Tyle że jeśli przed uruchomieniem, zaznaczę inną komórkę niż a1, to znów mam błąd.

Mógłbym zaznaczyć za pomocą makra a1, ale wolałbym inaczej wskazywać na komórki.

Pomoże nam słowo cells (komórki).

Cells wskazuje na komórkę, na podstawie liczb. Na przykład cells(1,1) wskazuje na a1 , a cells(1,2) na b1.

Już wiesz jak będzie wyglądał nasz finalny kod? Spróbuj napisać go samemu.

Jeśli udało Ci się napisać samemu to super
powinno to wyglądać mniej więcej tak :

```
For x = 1 to 10  
Cells(x,1) = x *2 & " ala ma kota"  
Next x
```