

Makropolecenia na arkuszu kalkulacyjnym.

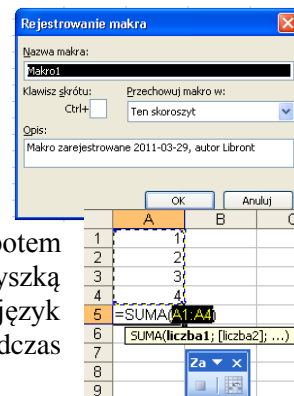
Język makropoleceń (język Visual Basic - VB) stanowi potężne narzędzie, za pomocą którego można zautomatyzować czynności wykonywane na arkuszu kalkulacyjnym. W połączeniu z możliwościami samego arkusza kalkulacyjnego daje to prawie nieograniczone możliwości tworzenia aplikacji użytkowych.

Zabezpieczenia

O sile VB może stanowić fakt, że znaczna ilość wirusów (tzw. makrowirusy) pisane są właśnie za pomocą języka VB i uruchamiane w środowisku MS Office (Word, Excel, itp.). Dlatego też aplikacje MS Office mają automatycznie zablokowaną możliwość uruchamiania makropoleceń. Ponieważ uniemożliwia to także uruchamianie naszych makr, więc konieczne będzie częściowe odbezpieczenie Excela. Z menu wybieramy: **Narzędzia – Makro – Zabezpieczenia** i w oknie ustawiamy poziom: **Średnie** - gdy otwieramy arkusz z makrami pojawia się okno akceptacji lub wybieramy poziom: **Niskie** – gdy jesteśmy w 100% pewni, co do uruchamianych makropoleceń.

Nagrywanie

Nawet jeśli nie znamy poleceń VB, bardzo łatwo nauczyć arkusz wykonywać zaplanowane operacje i odtwarzać je dowolną ilość razy. Aby „nagrać” wykonywane czynności należy z menu wybieramy: **Narzędzia – Makro – Zarejestruj nowe makro**. Pojawia się okienko rejestracji, w którym można określić nazwę makra, skrót, którym będziemy je uruchamiać i miejsce przechowywania. Możemy pozostawić domyślne ustawienia – wszystko można potem zmienić. Po wciśnięciu przycisku **OK** rozpoczynamy rejestrowanie poleceń. Każde kliknięcie myszką w arkusz, wybranie polecenia z menu czy wpisanie danych do komórek będzie zamieniane na język VB i wstawiane do specjalnego edytora (każda pomyłka również jest rejestrowana). Podczas rejestracji pojawia się mini pasek narzędziowy, za pomocą którego można zakończyć rejestrację.

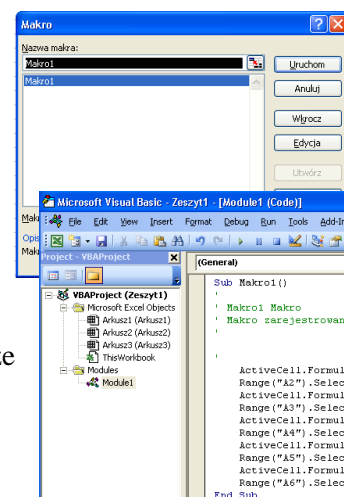


Edycja i Uruchamianie

Edytor makropoleceń wywołujemy wybierając z menu: **Narzędzia – Makro – Makra**. W okienku zaznaczamy odpowiednie makro i naciskamy przycisk **Edycja**. W podobny sposób można również uruchamiać wybrane makra – naciskając przycisk **Uruchom**

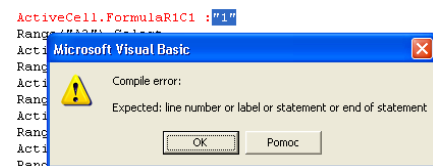
Za pomocą edytora makropoleceń można przeprowadzić najprostszą edycję kodu źródłowego. Edytor wspomaga nas kolorując kod, wyróżniając komentarze, wbudowane funkcje i pokazując błędy. Nasze makropolecenia pojawiają się w sekcji **Modules**, na stronie o nazwie **Module1**. Tak umieszczone procedury makropoleceń „widoczne” będą w całym skoroszycie (we wszystkich arkuszach). Można również pisać makra w konkretnych arkuszach – „widoczne” będą tylko w tym arkuszu.

Jeżeli wywołujemy edytor bezpośredni z nowootwartego arkusza (nie ma jeszcze zarejestrowanych makr) należy z menu edytora wybrać: **Insert – Module**.



Język VB

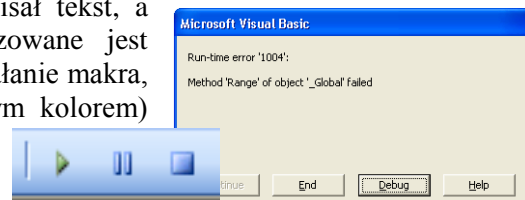
Każda zarejestrowana lub napisana własnoręcznie akcja lub ich zestaw tworzy procedurę rozpoczynającą się od słowa **Sub**, po którym występuje dowolna, jednowyrazowa nazwa i dwa nawiasy () (mogą zawierać dodatkowe parametry). Procedura kończy się słowami **End Sub**. Kolejne procedury można pisać pod spodem lub na nowych stronach (Module2, Module3,...). **Instrukcje języka VB** pisze się w osobnych wierszach. **Komentarze** oznacza się wstawiając na początku wiersza apostrof (zielony kolor wiersza). Wszystkie **poprawnie wpisane słowa kluczowe** (nazwy zarezerwowane dla VB) edytor zamienia wyrównując tekst i ustawiając w odpowiednich miejscach duże i małe litery.



Obsługa błędów

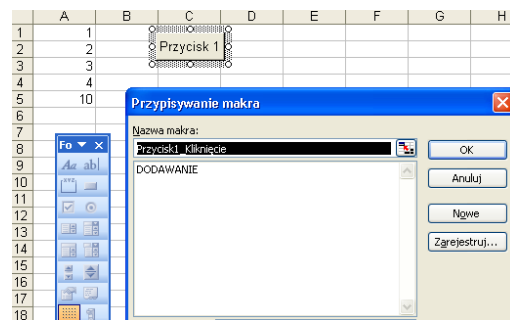
Niepoprawnie wpisane polecenia powodują automatyczny błąd przy próbie przejścia do nowego wiersza – wiersz oznaczony jest czerwonym kolorem, a fragment, od którego edytor „nie może zrozumieć” kodu źródłowego zaznaczony.

W trakcie działania makropolecenia może dojść do przerwania jego działania. Powodów może być kilka: sami zatrzymujemy makro klawiszami CTRL+PAUSE, niepoprawnie wpisano polecenie (np. Range(A1)=2, brak cudzysłowia) lub makro nie będzie w stanie wykonać polecenia (np. użytkownik wpisał tekst, a program oczekiwał liczby). Przerwanie wykonywania makra sygnalizowane jest okienkiem z napisem **Run-time error**. Wybranie przycisku **End** kończy działanie makra, a przycisku **Debug** spowoduje przejście do edytora i zaznaczenie (żółtym kolorem) polecenia, na którym wykonywanie makra zostało zatrzymane. Po poprawieniu błędów można uruchomić makro od tego miejsca lub przerwać je i uruchomić od nowa. Służą do tego przyciski paska narzędziowego.



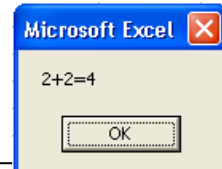
Tworzenie przycisku

Bardzo wygodnym sposobem uruchamiania makropoleczeń jest utworzenie przycisku na arkuszu i przypisanie do niego makropolecenia. Formatka przycisku znajduje się na pasku narzędziowym *Formularze*. Po narysowaniu przycisku w dowolnym miejscu arkusza pojawia się okienko z makropoleceniami – wybieramy odpowiednie makro i naciskamy przycisk **OK**. *Zmiana napisu na przycisku* - kliknięcie prawym przyciskiem myszy w przycisk i wybranie polecenia *Edytuj tekst*. *Przypisanie innego makra* do przycisku - *Przypisz makro*.



Okno komunikatu MsgBox

Wyświetlamy proste okienko z dowolnym komunikatem z jednym przyciskiem OK. Komunikaty mogą być prostymi lub złożonymi z fragmentów tekstami połączonymi z liczbami, datami, zmiennymi itp.



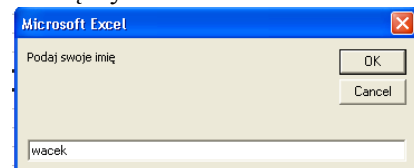
<code>MsgBox ("Ala ma kota")</code>	<i>napis</i>
<code>MsgBox "Ala ma kota"</code>	<i>można bez nawiasów</i>
<code>MsgBox "Ala " & "ma" & " kota"</code>	<i>identyczny komunikat, zwróć uwagę na odstępy i znak połączenia &</i>
<code>MsgBox "Ala" & " " & "ma" & " " & "kota"</code>	<i>jeszcze raz to samo</i>
<code>MsgBox "Ala" & chr(10) & "ma" & vbCrLf & "kota"</code>	<i>każdy wyraz w osobnym wierszu</i>
<code>MsgBox "2+2=" & 2+2</code>	<i>komunikat 2+2=4</i>

Polecenia Chr(10) lub vbCrLf przenoszą następny tekst do nowego wiersza.

Okno pobierania danych InputBox

Wyświetlamy okienko z komunikatem, polem tekstowym i dwoma przyciskami. Po wpisaniu tekstu lub liczby do pola tekstowego i naciśnięciu przycisku OK zostanie zawartość pola tekstowego przypisana do zmiennej. naciśnięcie przycisku Cancel – przypisanie do zmiennej tekstu pustego "". Parametry w okienku MsgBox można było wpisywać w nawiasach lub bez nich. W okienku InputBox nawiasy są niezbędne. Zmienne nie muszą być deklarowane. Wartość przechowywana w zmiennej wynik może być wykorzystana w dalszej części makra, np. do wyświetlenia w okienku MsgBox.

```
wynik = InputBox("Podaj swoje imię")
```



W sytuacji na obrazku obok – wciśnięcie klawisza OK spowoduje, że w zmiennej **wynik** znajdzie się napis "wacek", a przycisku Cancel – napis pusty ""

Komórki arkusza - czytanie danych z komórki i wpisanie danych do komórki

Range

Do obsługi komórek arkusza służą dwie funkcje RANGE i CELLS. Przy ich pomocy można pobierać zawartość komórki do zmiennej (lub np. bezpośrednio do okienka MsgBox) i wpisywać dowolne informacje do komórek. W poznanych wcześniej językach programowania, należało dość wyraźnie zadeklarować typy zmiennych. W języku VB nie jest to konieczne i sam program stara się rozpoznać, z jakim typem informacji ma do czynienia. Oczywiście pojawi się błąd, jeśli spróbujemy wykonać operację typu `Range("A1") = 2 * "Ala ma kota"`.

<code>Range("A1") = "Ala ma kota"</code>	<i>w komórce A1 tekst: Ala ma kota</i>
<code>Range("A2") = 2+2</code>	<i>w komórce A2 liczba 4</i>
<code>Range("A3") = #2011-03-28#</code>	<i>w komórce A3 data 2011-03-28</i>
<code>x = Range("A2")</code>	<i>w zmiennej x zawartość komórki A2 - 2</i>
<code>x=x+1</code>	<i>zwiększamy zmienną x o 1 (teraz 3)</i>
<code>Range("A5") = x</code>	<i>do komórki A5 wpisujemy liczbę 3</i>
<code>Range("A6") = Range("A6") + 1</code>	<i>zwiększamy zawartość komórki A6 o 1 (gdy komórka pusta, to pojawi się 1)</i>
<code>Range("A3") = Range("A3") + 3</code>	<i>2011-03-31</i>
<code>Range("A7") = "Następny dzień po " & Range("A3") & " to " & Range("A3")+ 1</code>	<i>w komórce A7 pojawi się tekst: Następny dzień po 2011-03-31 to 2011-04-01</i>

Cells

Identyczne operacje możemy wykonywać za pomocą funkcji Cells(w, k). W tym przypadku podajemy nie konkretny adres komórki lecz numer wiersza i kolumny, np. Cells(1,1) oznacza komórkę A1, Cells(3,2) – komórka B3.

```
Cells(6,1) = Cells(6,1) + 1
```

zwiększenie zawartości komórki A6 o 1

Ćwiczenie 1 – rejestrowanie dodawania

- Włącz rejestrator makropoleceń
- Wpisz do komórek od A1 do A4 liczby 1,2,3,4
- Do komórki A5 wpisz formułę =SUMA(A1:A4)
- Zakończ rejestrowanie – BARDZO WAŻNE

UWAGA – podczas pracy nie wykonuj żadnych niepotrzebnych operacji

- Wejdź do edytora makropoleceń i zmień nazwę makra na DODAWANIE
- Przejdź do arkusza kalkulacyjnego i usuń wszystko z tych pięciu komórek
- Uruchom jeszcze raz polecenie DODAWANIE

Ćwiczenie 2 – rejestrowanie czyszczenia

- Włącz rejestrator makropoleceń
- Wyczyść komórki od A1 do A5 (kolejno lub zaznacz wcześniej wszystkie)
- Zakończ rejestrowanie
- Przejdź do edytora i zmień nazwę makra na CZYŚĆ.

Ćwiczenie 3 – przyciski dodawania i czyszczenia

- Przejdź do arkusza
- Utwórz nowy przycisk i przypisz do niego makro DODAWANIE
- Zmień napis na przycisku na DODAWANIE
- Utwórz nowy przycisk i przypisz do niego makro CZYŚĆ
- Zmień napis na przycisku na CZYŚĆ

Ćwiczenie 4 – rejestrowanie rysowania

- Włącz rejestrator makropoleceń
- Narysuj w dowolnym miejscu czerwony prostokąt
- Zakończ rejestrowanie
- Utwórz nowy przycisk i przypisz do niego nowe makro
- Przejdź do edytora
- Zmieniaj liczby w poleceniu rysującym kwadrat i sprawdź, które odpowiadają za położenie wymiary i kolor

Ćwiczenie 5 – równanie kwadratowe

- Załóżmy, że w komórkach B1, B2 i B3 będą znajdować się współczynniki równania kwadratowego wpisane przez użytkownika, a komórkach B4, B5, B6 pojawią się wyliczone wartości delty i pierwiastków równania kwadratowego.
- Makropolecenie powinno wczytać dane z tych komórek do zmiennych (nazwy np. a, b, c) za pomocą Range lub Cells.
- Makro powinno wyliczyć deltę i pierwiastki równania, a wyliczone wartości przypisać do nowych zmiennych (np. delta, x1 i x2).
- Zawartość tych zmiennych wstawić do komórek B4, B5 i B6.
- Utwórz odpowiedni przycisk i przypisz do niego to makropolecenie.

	A	B
1	A	2
2	B	-5
3	C	1
4	delta	17
5	x1	0,219224
6	x2	2,280776

Ćwiczenie 6 – prostopadłościan

Napisać program obliczający pole powierzchni i objętość prostopadłościanu, którego boki mają długość a , b , c . Dane a , b , c wprowadzić z okienek InputBox. Wyniki umieścić w okienkach MsgBox. Podobnie jak w poprzednim przykładzie – działać na zmiennych.

Ćwiczenie 7 – długość prostej

Dane są dwa punkty $A(x1,y1)$ i $B(x2,y2)$. Obliczyć długość odcinka AB oraz współrzędne jego środka. Dane wprowadzić z okienek InputBox. Wyniki umieścić w okienkach MsgBox

Ćwiczenie 8 – potęgowanie

W okienku InputBox wczytujemy liczbę, a w okienku MsgBox wyświetlamy pierwiastek kwadratowy i kwadrat tej liczby w postaci (trzy wiersze – użyj polecenia Chr(10) lub vbCrLf):

Wpisano liczbę: 9

Pierwiastek kwadratowy 9 to: $9^{(1/2)} = 3$

Kwadrat liczby 9 to: $9^2 = 81$

Przykładowe rozwiązania zadań w arkuszu *arkusz – makropolecenia (01).xls*