

# VBA



## Zmienne

służą do przechowywania danych określonego typu w pamięci komputera.

## Typy danych

Byte	zmienna bitowa
Integer	liczba całkowita
Double	liczba zmiennoprzecinkowa
String	napis
Boolean	zmienna logiczna true/false
Date	data i godzina

## Zasięg zmiennych

Zmienna lokalna - wewnątrz procedury (pomiędzy poleceniami sub/function, a End Sub/End Function)

Zmienna modułowa - przed pierwszą deklaracją procedury w module, polecenie Dim lub Private

Zmienna globalna - w module standardowym VBA, przed pierwszą procedurą modułu (poza wszystkimi procedurami), polecenie Public

## Funkcje operujące na ciągach znaków

Str	Zwraca daną typu String (konwersja liczby na napis)
Int	Zwraca daną typu Integer (konwersja napisu na liczbę całkowitą)
LCase	Zwraca łańcuch typu String przekształcony na małe litery
UCase	Zwraca łańcuch typu String przekształcony na wielkie litery
Len	Zwraca daną typu Long określającą liczbę znaków w łańcuchu znaków
LTrim	Zwraca daną typu String zawierającą kopię łańcucha bez początkowych spacji
RTrim	Zwraca daną typu String zawierającą kopię łańcucha bez końcowych spacji
Trim	Zwraca daną typu String zawierającą kopię łańcucha bez początkowych i końcowych spacji
Replace	Zwraca łańcuch znaków, w którym określony podłańcuch jest zastępowany innym łańcuchem znaków określoną ilość razy
Left	Zwraca daną typu String zawierającą określoną liczbę znaków z lewej strony łańcucha znaków

Right  
Zwraca daną typu String zawierającą określoną liczbę znaków z prawej strony łańcucha znaków

Mid  
Zwraca daną typu String zawierającą określoną liczbę znaków wewnątrz ciągu

## Komentarze

Komentarze pełnią funkcję opisu kodu i nie są one wykonywane. Komentarz rozpoczyna się znakiem apostrofu

Przykład:

```
liczba = 0 'to jest komentarz  
napis = ""to nie jest komentarz""
```

## Operatory matematyczne

^	potęgowanie
Mod	reszta z dzielenia
*	mnożenie
/	dzielenie
\	dzielenie całkowite
+	dodawanie
-	odejmowanie

## Operatory porównania

<	mniejsze
<=	mniejsze lub równe
>	większe
>=	większe lub równe
=	równe
<>	różne
Like	zgodność ciągu znaków z wzorcem

## Operatory konkatencji (łączenia)

& - konkatencja dwóch wyrażeń, której zwracany wynik jest ciągiem znaków

Przykład:

```
Napis = "Jan " & "Kowalski" 'daje  
wynik "Jan Kowalski"
```

## Operatory logiczne

	koniunkcja (iloczyn logiczny) prawda And prawda = prawda, prawda And fałsz = fałsz, fałsz And prawda = fałsz, fałsz And fałsz = fałsz
Or	alternatywa (suma logiczna) prawda Or prawda = prawda, prawda Or fałsz = prawda, fałsz Or prawda = prawda, fałsz Or fałsz = fałsz
Not	negacja Not prawda = fałsz, Not fałsz = prawda

## Instrukcje warunkowe

If *warunek* Then  
*Instrukcje*  
[ElseIf *warunek*  
Then  
*Instrukcje*]  
[Else  
*Instrukcje*]  
End If

Warunkowo wykonuje grupę instrukcji zależnie od spełnienia warunku. Każdy warunek może przyjmować dwie wartości: prawda (True) lub fałsz (False). Zależnie od wartości wykonywany jest określony blok instrukcji. Klauzule ElseIf i Else są opcjonalne

Select Case *wyrażenie*  
[Case *wartość\_wyrażenia*  
*Instrukcje*]  
[Case Else  
*instrukcje\_domyślne*]  
End Select

Wykonanie jednego z kilku bloków instrukcji zależnie od wartości podanego wyrażenia. *Wartość\_wyrażenia* może przyjmować następujące formy:  
Case 1, 5, 11 – wartości  
Case 1 To 10 – zakres wartości  
Case Is < 10 – wyrażenie porównawcze  
Case 1 To 10, Is > 20 – można łączyć kilka form

## Pętle

For *licznik* = *start* To *koniec*  
[Step *krok*]  
*Instrukcje*

Powtarza blok instrukcji określoną liczbę razy

Next *licznik*

Do [While | Until] *warunek*  
*Instrukcje*  
Loop

Powtarza blok instrukcji, dopóki warunek While jest spełniony lub warunek Until nie jest spełniony

## Procedura typu Sub

nie zwraca żadnej wartości. Wcześniejsze zakończenie procedury następuje poprzez użycie Exit Sub.

Sub *NazwaProcedury*([*Lista\_argumentów*])  
End Sub 'koniec procedury

Wywołanie procedury Sub  
*NazwaProcedury* *argument1*, *argument2* ...  
Call *NazwaProcedury*(*argument1*, *argument2* ...)

Wywołanie procedury Sub z innego modułu  
*nazwaModułu*.*nazwaProcedury* *argument1*, *argument2* ...  
Call *nazwaModułu*.*nazwaProcedury*(*argument1*, *argument2* ...)

## Procedura typu Function

zwana również funkcją, wykonuje określone obliczenia oraz zwraca pojedynczą wartość. Mogą być stosowane w modułach i formułach arkuszy. Wcześniejsze zakończenie procedury następuje poprzez użycie Exit Function. Wartość zwracaną przez funkcję należy przypisać

nazwie funkcji (w trakcie procedury można wielokrotnie dokonywać takiej operacji).  
Function *NazwaFunkcji* ([*Parametry*])

End Function

Wywoływanie funkcji  
Procedury typu Function można wywoływać w formule arkuszy oraz z innych procedur.  
*wynik* = *nazwaFunkcji*(*argument1*, *argument2* ...)

Wywołanie procedury z innego modułu  
*wynik* = *nazwaModułu*.*nazwaFunkcji*(*argument1*, *argument2* ...)

## Funkcje dotyczące daty i czasu

Weekday Zwraca daną typu Integer zawierającą liczbę reprezentującą dzień tygodnia

DateValue Zwraca daną typu Date (konwersja napisu na datę)

TimeValue Zwraca daną typu Date (konwersja napisu na czas)

Year Zwraca daną typu Integer reprezentującą rok

Month Zwraca daną typu Integer reprezentującą miesiąc roku

Day Zwraca daną typu Integer reprezentującą dzień miesiąca

Hour Zwraca daną typu Integer reprezentującą godzinę dnia

Minute Zwraca daną typu Integer reprezentującą minutę godziny

Second Zwraca daną typu Integer reprezentującą sekundę minuty

## Deklarowanie zmiennych

Dim *nazwaZmiennej* As *typZmiennej* = *wyrażenie*

## Stałe

Nazwy zastępujące wartości niezienne w kodzie programu, nazwy o określonym znaczeniu. Deklaracja stałych następuje przy użyciu instrukcji Const.

Const *nazwaStalej* As *typDanej* = *wyrażenie*

