VBA dla Excela 2002/2003. Leksykon kieszonkowy

Autor: Mirosław Lewandowski
ISBN: 83-7361-377-3
Format: B6, stron: 144

Ta niewielka rozmiarami książka jest niezwykle cenną pomocą dla osób, które chcą bez wgłębiania się w niuanse Visual Basic for Applications tworzyć ad hoc skrypty w tym języku, rozszerzając możliwości programu Microsoft Excel.
W zwięzłej i skondensowanej formie znajdziesz tu wybrane, najważniejsze informacje o programowaniu w VBA i dziesiątki drobnych wskazówek, które pomogą Ci osiągnąć cele, które sobie postawiłeś. Nie zawsze trzeba sięgać po podręcznik wyjaśniający wszystko od podstaw. Jeżeli nie czujesz takiej potrzeby, wystarczy Ci książka, którą właśnie trzymasz w ręku.

Omówiono między innymi

- Stałe, zmienne i tablice w VBA
- Obiekty i metody; w tym wybrane obiekty MS Office
- Funkcje Visual Basica
- Interakcję programów z użytkownikiem
- Instrukcje Visual Basica
- Obsługę błędów w VBA
- Procedury zdarzeniowe dla obiektów Excela
- Formularze i związane z nimi procedury

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 3. Elementy Visual Basica

Funkcje i operatory matematyczne

Funkcje trygonometryczne

Do wyboru mamy funkcje:

- `Atn` — Arcus tangens,
- `Cos` — Cosinus,
- `Sin` — Sinus,
- `Tan` — Tangens.

Składnia wszystkich jest taka sama:

```
Funkcja(Wartość)
```

Aby otrzymać wartość funkcji cotangens, należy zastosować funkcję 1/Tan.

Wartość Pi możesz obliczyć na dwa sposoby:

- w VBA jako ArcusTangens:

```
Pi = 4 * Atn(1)
```

- lub korzystając z funkcji arkuszowej `Pi`:

```
pi = WorksheetFunction.Pi
```

Exp i Log

Log zwraca wartość logarytmu naturalnego danej liczby. Podstawą logarytmów naturalnych jest stała e=2,71828182845904.

`Exp` jest odwrotnością funkcji `Log` — zwraca wartość liczby e podniesioną do wskazanej potęgi.

Składnia:

```
Exp(Wykładnik)
```

- *Wykładnik* — wykładnik potęgi,

```
Log(Liczba)
```

- *Liczba* — liczba rzeczywista dodatnia, której logarytm należy obliczyć.

Sqr

Zwraca pierwiastek kwadratowy podanego argumentu.

Składnia:

```
Sqr(Argument)
```

- *Argument* — liczba rzeczywista większa od 0.

Randomize, i Rnd

`Randomize` służy do zainicjowania generatora liczb losowych.

Składnia:

```
Randomize(Baza)
```

- *Baza* — argument opcjonalny — wartość początkowa do obliczenia zbioru liczb pseudolosowych. Jeżeli go pominiesz, zostanie on ustalony na podstawie wskazań zegara systemowego, co dodatkowo korzystnie wpłynie na losowane liczby.

`Rnd` generuje liczbę losową z zakresu od 0 do <1.

Składnia:

```
Rnd(Liczba)
```

- *Liczba* — argument opcjonalny;
- jeżeli *Liczba* = 0, funkcja zwróci ostatnio wygenerowaną liczbę;
- jeżeli *Liczba* <0, funkcja za każdym razem zwróci tę samą, raz wygenerowaną wartość;
- jeżeli pominiesz argument lub *Liczba* >0, funkcja zwróci kolejną liczbę ze zbioru liczb losowych.

Wartość argumentów *Baza* i *Liczba* nie ma znaczenia, jeżeli zależy Ci na losowym generowaniu liczb. Jednakże za ich pomocą możesz wpłynąć na powtórzenie generowania tego samego zestawu. Jeśli więc chcesz, aby liczby losowe zaczęły powtarzać się w tej samej kolejności przed zainicjowaniem generatora, wywołaj funkcję Rnd z parametrem ujemnym, a następnie zainicjuj generator liczb losowych. Wyjaśni to poniższy przykład:

```
Sub pseudolosowa()
'pierwsza inicjacja
 Rnd (-1)
 Randomize 2
 For a = 1 To 20
 Cells(a, 1) = Rnd
 Next
'druga inicjacja
 Rnd (-1)
 Randomize 2
 For a = 1 To 20
 Cells(a, 2) = Rnd
 Next
End Sub
```

W przykładzie otrzymamy dwie kolumny z wygenerowanymi losowo liczbami z zakresu 0 do 1. Losowo, lecz w tej samej kolejności.

Abs

Oblicza wartość bezwzględną (moduł) podanej liczby, czyli odcina znak minus, jeżeli występuje.

Składnia:

```
Abs(Liczba)
```

- *Liczba* — dowolna liczba rzeczywista.

Sgn

Zwraca wartość w zależności od znaku podanego argumentu.

Składnia:

```
Sgn(Argument)
```

- *Argument* — dowolna liczba rzeczywista.

Funkcja zwraca następujące wartości:

`-1` — gdy argument jest mniejszy od zero,

`0` — gdy argument jest równy zero,

`1` — gdy argument jest większy od zero.

Fix, Int

Zwracają część całkowitą argumentu.

Składnia:

```
Fix(Argument)
Int(Argument)
```

- *Argument* — dowolna liczba rzeczywista.

W zakresie liczb dodatnich funkcje odcinają część ułamkową argumentu. Różnice w działaniu są widoczne podczas działań na

liczbach ujemnych. `Int` zaokrągla argument w dół, podczas gdy `Fix` — w górę.

Przykład:

`Int(3.2)`	Da wynik 3
`Fix(3.2)`	Da wynik 3
`Int(-3.2)`	Da wynik –4
`Fix(-3.2)`	Da wynik –3

Operatory matematyczne

Znak	Opis	Składnia — przykład użycia
^	Znak potęgowania	wynik = liczba^wykładnik
- +	Znaki odejmowania i dodawania	wynik = składnik + składnik
* /	Znaki mnożenia i dzielenia	wynik = dzielna/dzielnik
\	Zwraca część całkowitą z wyniku dzielenia. Dodatkowo dzielna i dzielnik przed wykonaniem obliczeń zostaną pozbawione części ułamkowej	wynik = dzielna\dzielnik
Mod	Zwraca resztę z dzielenia	reszta = dzielna Mod dzielnik
&	Służy do łączenia dwóch ciągów znaków	wynik = "łańcuch1"&"łańcuch2"

Round

Zwraca liczbę zaokrągloną do zadanej dokładności.

Składnia:

`Round (Liczba, IleMiejsc)`

- *`Liczba`* — wymagany — dowolna liczba rzeczywista poddana zaokrągleniu;
- *`IleMiejsc`* — opcjonalny — wskazuje, z jaką dokładnością (do ilu miejsc po przecinku) należy zaokrąglić liczbę. Jeżeli pominiesz ten parametr, funkcja zwróci liczbę całkowitą.

Data i czas

Hour, Minute, Second

Funkcje zwracają godzinę, minutę lub sekundę z podanego argumentu. Argumentem może być liczba w postaci dziesiętnej lub w formacie czasu.

Przykład:

```
Hour(0.593888889)
Hour(#2:15:12 PM#)
```

W obu powyższych przypadkach funkcja zwróci liczbę 14, bowiem obydwa argumenty przedstawiają tę samą godzinę. Analogicznie:

```
Minute(0.593888889)
```

da wynik 15. Natomiast:

```
Second(0.593888889)
```

da wynik 12

Day, Month, Year

`Day` zwraca liczbę o wartości od 1 do 31 reprezentującą kolejny dzień miesiąca.

`Month` zwraca liczbę w zakresie od 1 do 12 reprezentującą miesiąc roku z podanej daty.

`Year` zwraca rok z podanej daty.

Składnia:

```
Day(data)
Month(data)
Year(data)
```

gdzie `data` to wyrażenie reprezentujące datę.

Weekday

Funkcja zwraca wartość liczbową (od 0 do 7) reprezentującą dzień tygodnia wskazanej daty.

Składnia:

`Weekday(Data, 1dzieńTygodnia)`

- *Data* — wymagany;
- *1dzieńTygodnia* — opcjonalny — wskazuje pierwszy dzień tygodnia.

Przykład:

`Weekday(data,2)` zwróci wartość 1, jeżeli rozpatrywany dzień będzie poniedziałkiem.

`Weekday(data,4)` zwróci wartość 1 dla środy, 2 dla czwartku i tak dalej.

Domyślną wartością parametru pierwszy jest 1 (czyli niedziela).

TimeSerial

Zwraca w wyniku czas.

Składnia:

`TimeSerial(Godzina, Minuta, Sekunda)`

- *Godzina*, *Minuta*, *Sekunda* — wymagane — dowolne dodatnie liczby całkowite.

Przykład:

`TimeSerial(2, 34, 7)`	Da w wyniku godzinę 2:34:07

TimeValue

Konwertuje ciąg znaków o ustalonej składni na zmienną zawierającą czas.

Przykład:

`TimeValue("4:35:17 PM")` da w wyniku zmienną typu `Date` wskazującą czas 16:35:17.

DateSerial

Zwraca w wyniku datę.

Składnia:

```
DateSerial(Rok, Miesiąc, Dzień)
```

- *Rok*, *Miesiąc*, *Dzień* — wymagane — dowolne liczby całkowite.

Przykład:

`DateSerial(0, 4, 7)`	Da w wyniku datę 07.04.2000 roku
`DateSerial(99, 4, 7)`	Da w wyniku datę 07.04.1999 roku
`DateSerial(100, 4, 7)`	Da w wyniku datę 07.04.100 roku

Warto stosować pełny (czterocyfrowy) zapis roku, aby uniknąć pomyłek pokazanych powyżej.

DateValue

Konwertuje ciąg znaków o ustalonej składni na zmienną typu `Date` zawierającą datę.

Przykłady:

```
DateValue("luty 3 3002")
DateValue("3 luty 3002")
```

W powyższych linijkach zostanie obliczona data 3.02.3002 roku.

```
DateValue("3 2 3002")
DateValue("3,2,3002")
```

Po wykonaniu powyższych poleceń program zwróci wartość 2.03.3002 roku.

Poniższy zapis spowoduje błąd:

```
DateValue(3, 2, 3002)
```

VBA obsługuje daty z zakresu od 1.01.100 do 31.12.9999 roku i wyrażenia zawierające takie wartości mogą zostać podstawione jako argument funkcji `DateSerial`.

DateAdd

Dodaje do podanej daty określony interwał czasowy.

Składnia:

```
DateAdd(Interwał, Ilość, Data)
```

- *Interwał* — wymagany — podaje, jaki przedział czasowy będzie dodany do daty.

Możliwe wartości:

yyyy	Rok
q	Kwartał
m	Miesiąc
y	Dzień roku
d	Dzień
w	Dzień tygodnia
ww	Tydzień
h	Godzina
n	Minuta
s	Sekunda

Na potrzeby funkcji DateAdd parametry y, d, i w oznaczają zawsze dodanie dnia do wskazanej daty. Jednak przy innych funkcjach daty i czasu parametry te mają już różne znaczenia.

- *Ilość* — wymagany — wskazuje, ile interwałów czasowych ma być dodanych;
- *Data* — wymagany — data bazowa.

Przykład:

```
data = DateSerial(2, 4, 7) 'tworzymy datę 07.04.2002
nowa_data1 = DateAdd("n", 3, data)
nowa_data2 = DateAdd("d", 3, data)
nowa_data3 = DateAdd("q", 3, data)
nowa_data4 = DateAdd("ww", 3, data)
nowa_data5 = DateAdd("yyyy", 3, data)
```

W wyniku działania powyższego kodu zmienne przyjmą następujące wartości:

nowa_data1	2002-04-07 00:03
nowa_data2	2002-04-10
nowa_data3	2003-01-07
nowa_data4	2002-04-28
nowa_data5	2005-04-07

DateDiff

Zwraca różnicę między podanymi datami.

Składnia:

```
DateDiff(Interwał, Data1, Data2,1dzieńTygodnia, 1TydzieńRoku)
```

- *Interwał* — wymagany — patrz funkcja DateAdd;
- *Data1*, *Data2* — wymagane — daty, między którymi zostanie obliczona różnica;

- *1dzieńTygodnia* — opcjonalny — stała wskazująca początek tygodnia. Możliwe są wartości od 0 (niedziela) do 7 (sobota) lub stałe z kolekcji vbDayOfWeek;
- *1TydzieńRoku* — opcjonalny — stała wskazująca, w jaki sposób ma zostać wskazany pierwszy tydzień roku.

Możliwe wartości:

vbUseSystem lub 0	Używa ustawień systemowych
vbFirstJan1 lub 1	Pierwszym jest tydzień zawierający dzień 1 stycznia
vbFirstFourDays lub 2	Pierwszym jest tydzień, w którym przynajmniej cztery dni należą do nowego roku
vbFirstFullWeek lub 3	Pierwszy pełny tydzień roku

DatePart

Oblicza, w jakiej części interwału czasowego mieści się podana data.

Składnia:

```
DatePart(Interwał, Data, 1dzieńTygodnia, 1TydzieńRoku)
```

Parametry zostały opisane przy funkcjach DateDiff i DateAdd.

Przykład:

```
data = DateSerial(2, 4, 7) 'tworzymy datę 07.04.2002
nowa_data1 = DatePart("w", data)
nowa_data2 = DatePart("y", data)
nowa_data3 = DatePart("q", data)
nowa_data4 = DatePart("ww", data)
nowa_data5 = DatePart("yyyy", data)
```

W wyniku działania powyższego kodu, zmienne nowa_data przyjmą następujące wartości:

nowa_data1	1	— wskazana data to niedziela
nowa_data2	97	— wskazana data to 97. dzień roku
nowa_data3	2	— kwiecień jest w drugim kwartale
nowa_data4	15	— wskazaną datę obejmuje 15. tydzień roku
nowa_data5	2002	— wskazaną datę obejmuje rok 2002

Date, Now, Time

- `Date` zwraca dzisiejszą datę;
- `Time` zwraca aktualny czas;
- `Now` zwraca wyrażenie w postaci dzisiejszej daty i aktualnego czasu.

Wartości są obliczane na podstawie z zegara systemowego.

Składnia:

```
zmienna = Date
zmienna = Time
zmienna = Now
```

Funkcje bezparametrowe.

Timer

Wskazuje, ile sekund (wraz z ułamkami) upłynęło od północy. Funkcja bezparametrowa.

Składnia:

```
zmienna = Timer
```

MonthName

Podaje (po polsku!) nazwę miesiąca.

Składnia:

```
MonthName(Numer, Skrócona)
```

- *Numer* — wymagany — podaje numer miesiąca;
- *Skrócona* — opcjonalny — jeżeli wprowadzisz wartość `True`, to nazwa miesiąca będzie podana w formie skróconej (na przykład *mar* zamiast *marzec*). Domyślna wartość to `False`.

WeekdayName

Podaje (po polsku) nazwę dnia tygodnia.

Składnia:

```
WeekdayName(Dzień, Skrócona, 1dzieńTygodnia)
```

- *Dzień* — wymagany — numer dnia;
- *Skrócona* — opcjonalny. Patrz funkcja `MonthName`;
- *1dzieńTygodnia* — opcjonalny — wskazuje pierwszy dzień tygodnia. Patrz funkcje `WeekDay` i `DateDiff`.

Calendar

Właściwość, która zwraca lub ustawia rodzaj używanego kalendarza w Twoim projekcie.

Składnia:

```
Calendar = jaki
```

Możliwe są dwie wartości parametru:

`vbCalGreg` lub `0`	Kalendarz gregoriański
`vbCalHijri` lub `1`	Hidżra — kalendarz księżycowy używany w krajach islamskich

Interakcja z użytkownikiem

MsgBox

Wyświetla okno komunikatu. Może także służyć do pobierania danych od użytkownika.

Składnia:

`MsgBox(Tekst, Przyciski, Tytuł, PlikPomocy, Kontekst)`

- *Tekst* — wymagany — komunikat, który zostanie wyświetlony — może nim być ciąg do 1024 znaków lub zmienna;
- *Przyciski* — opcjonalny — niesie informację o tym, jakie przyciski będą wyświetlone w oknie oraz jaki będzie typ komunikatu. Z typem komunikatu wiąże się wyświetlana w oknie ikona i efekty dźwiękowe (jeżeli użytkownik z nich korzysta).

Wartości przycisków okna:

`vbOKOnly` lub 0	Wartość domyślna — tylko przycisk *OK*
`vbOKCancel` lub 1	Przyciski *OK* i *Anuluj*
`vbAbortRetryIgnore` lub 2	Przyciski *Przerwij*, *Ponów próbę*, *Ignoruj*
`vbYesNoCancel` lub 3	*Tak*, *Nie*, *Anuluj*
`vbYesNo` lub 4	*Tak*, *Nie*
`vbRetryCancel` lub 5	*Ponów próbę*, *Anuluj*
`vbMsgBoxHelpButton` lub 16384	Dodatkowo przycisk *Pomoc*

Wartości typu komunikatu:

`vbCritical` lub 16	Zatrzymanie krytyczne
`vbQuestion` lub 32	Pytanie
`vbExclamation` lub 48	Ostrzeżenie
`vbInformation` lub 64	Informacja

`vbMsgBoxRight` lub 524288	Tekst jest wyrównany do prawej
`vbMsgBoxRtlReading` lub 1048576	Arabski układ okna (od prawej do lewej)

Odpowiednią wartość parametru `Przyciski` oblicza się przez dodanie do siebie wartości stałych (można podać składniki rozdzielone znakiem + lub ich sumę) albo podanie ich nazw rozdzielonych znakiem +.

- *`Tytuł`* — opcjonalny — komunikat, który będzie widoczny na pasku tytułu (jeżeli go pominiesz, zostanie tam wyświetlona nazwa „Microsoft Excel”);
- *`PlikPomocy`*, *`Kontekst`* — plik pomocy i miejsce w nim, do którego prowadzić będzie łącze po kliknięciu przycisku *Pomoc*.

Funkcja `MsgBox` może zwrócić wartości w zależności od akcji podjętej przez użytkownika:

`vbOK` lub 1	Kliknięto przycisk *OK*
`vbCancel` lub 2	Kliknięto przycisk *Anuluj*
`vbAbort` lub 3	Kliknięto przycisk *Przerwij*
`vbRetry` lub 4	Kliknięto przycisk *Ponów Próbę*
`vbIgnore` lub 5	Kliknięto przycisk *Ignoruj*
`vbYes` lub 6	Kliknięto przycisk *Tak*
`vbNo` lub 7	Kliknięto przycisk *Nie*

InputBox

Wynikiem wykonania tej funkcji jest wartość typu `String` wpisana przez użytkownika w oknie dialogowym.

Składnia:

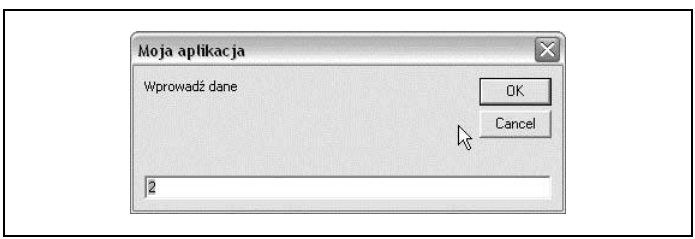
```
InputBox(Komunikat, Tytuł, Domyślna, x, y, PlikPomocy,
↳Kontekst)
```

- *Komunikat* — wymagany — parę słów zachęty dla użytkownika — będą one wyświetlone w oknie komunikatu;
- *Tytuł* — opcja — komunikat który będzie widoczny na pasku tytułu — jeżeli go pominiesz, zostanie tam wyświetlona nazwa „Microsoft Excel";
- *Domyślna* — opcja — zawiera wartość domyślną wprowadzanej zmiennej — będzie wyświetlana w miejscu wprowadzania danych (jeżeli pominiesz ten parametr, Excel nie wyświetli żadnej wartości w oknie);
- *x*, *y* — opcja — współrzędne (w pikselach) lewego górnego narożnika okna dialogowego względem lewego górnego narożnika ekranu;
- *PlikPomocy*, *Kontekst* — plik pomocy i miejsce w nim, do którego prowadzić będzie łącze po kliknięciu przycisku *Pomoc*.

Przykład:

Efektem wykonania poniższego kodu będzie okno dialogowe pokazane na rysunku 3.1. Jeżeli użytkownik nie wprowadzi żadnej wartości i kliknie *OK*, zmiennej `a` zostanie przypisana wartość 2. Jeżeli wybierze przycisk *Cancel*, funkcja zwróci wartość ciągu zerowej długości.

```
a = InputBox("Wprowadź dane", "Moja aplikacja", 2)
```



Rysunek 3.1. Okno dialogowe wyświetlone za pomocą funkcji InputBox

Funkcje logiczne

VBA oferuje pełną gamę ogólnie znanych operatorów logicznych:

`Not`, `And`, `Or`, `Xor`, `Eqv`, `Imp`.

Wszystkich oprócz operatora `Not` możemy używać w taki sam sposób:

```
wynik = wartość1 operator wartość2
```

gdzie `wartość1` i `wartość2` to wyrażenia, na których dokonuje się operacji.

Operator `Not` ma jeszcze łatwiejszą składnię:

```
wynik = Not argument
```

czego wynikiem będzie oczywiście odwrotność podanego argumentu.