

Techniki programowania



Instrukcja warunkowa i wyboru

projektowanie algorytmów

instrukcje sterujące VBA

instrukcja warunkowa

operatory relacyjne i logiczne

instrukcja wyboru

Projektowanie algorytmów

Algorytm – przepis postępowania prowadzący do rozwiązania określonego zadania; zbiór poleceń określających sposób przetwarzania zbioru danych ze wskazaniem kolejności w jakiej mogą być wykonane.

Realizacja każdego programu powinna być poprzedzona procesem projektowania, którego istotnym elementem jest zaplanowanie sposobu przetwarzania danych. Algorytm stanowi uniwersalny język zapisu, niezależny od docelowego narzędzia programowania.

Podstawowe symbole



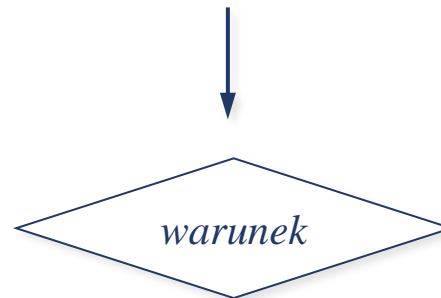
początek
algorytmu



koniec
algorytmu



operacja lub
proces



przejdźcie do
następnej operacji

blok decyzyjny

operacja
wejścia-wyjścia

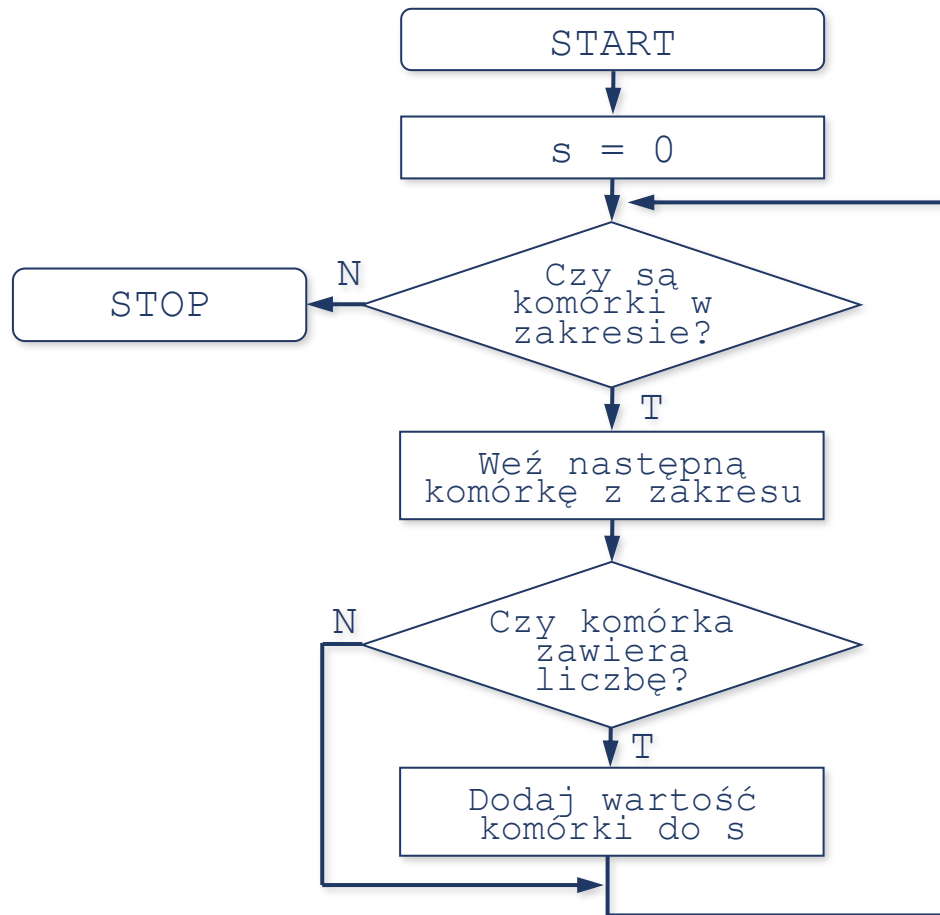
Zadanie: zaprojektować algorytm obliczający sumę wartości liczbowych w podanym zakresie arkusza.

	A	B	C
1		1	1
2		2	2
3	a		a
4	01.03.2022		01.03.2022
5	b		b
6		3	3
7		4	4
8		44631	10

=SUMA(A1:A7)

=SumaLicz(C1:C7)

Uwaga: standardowa funkcja SUMA traktuje daty jak wartości liczbowe.



Instrukcja sterująca – element języka programowania, który służy do określenia kolejności wykonania instrukcji zawartych w kodzie programu.

Instrukcje sterujące w VBA

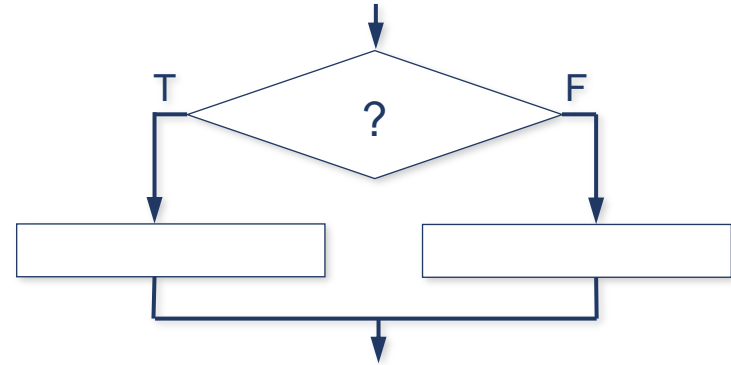
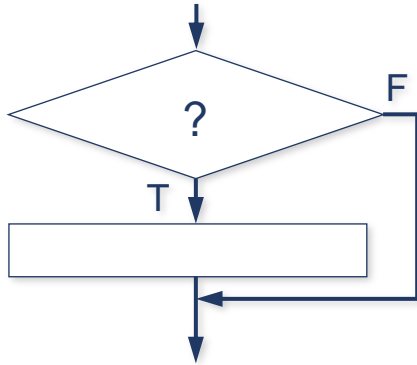
- **Instrukcja warunkowa** (**If-Then**, **If-Then-Else**) – wprowadza rozgałęzienie w kodzie programu, tworząc alternatywne sekwencje instrukcji.
- **Instrukcja wyboru** (**Select Case**) – wprowadza rozgałęzienie w kodzie programu tworząc dowolną liczbę alternatywnych sekwencji instrukcji.
- **Instrukcje iteracyjne (pętli)** (**For-Each**, **For-Next**, **Do-Loop**) – umożliwiają cykliczne powtarzanie sekwencji instrukcji.

Dodatkowo

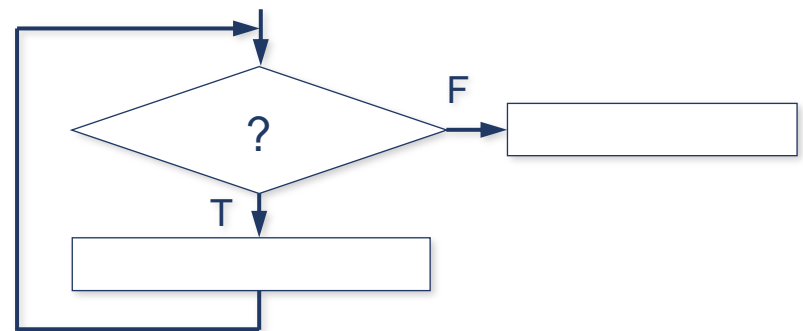
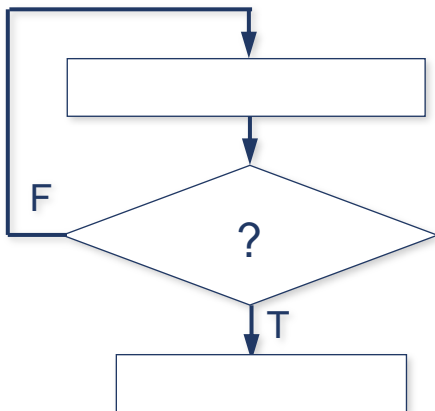
- **Funkcje wyboru wartości z listy** (**Choose**, **Switch**) – umożliwiają wybór wartości z predefiniowanej listy, zastępują szereg instrukcji warunkowych lub instrukcję wyboru.

Instrukcje sterujące w algorytmach

Instrukcje warunkowe



Instrukcje iteracyjne (pętli)



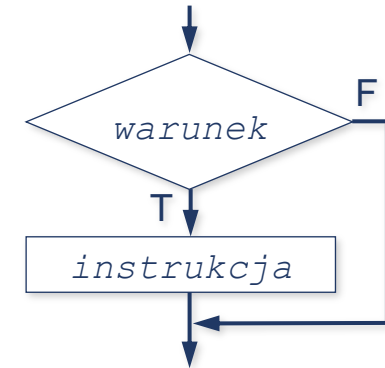
Instrukcja warunkowa

Jeżeli *warunek* jest prawdziwy wykonaj *instrukcję* (grupę instrukcji).

```
If warunek Then instrukcja
```

```
If warunek Then  
  instrukcja1  
  ...  
  instrukcjaN
```

```
End If
```



Jeżeli *warunek* jest prawdziwy wykonaj *instrukcję1* (pierwszą grupę instrukcji) w przeciwnym wypadku wykonaj *instrukcję2* (drugą grupę instrukcji).

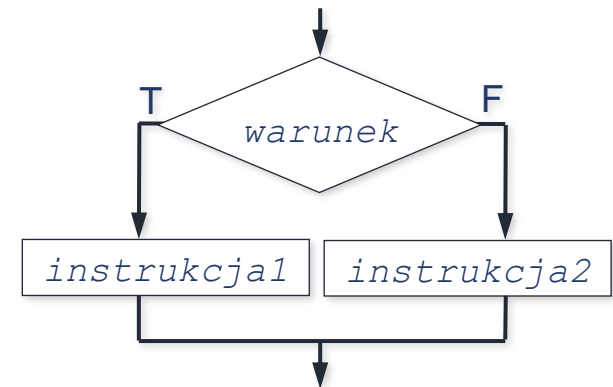
```
If warunek Then instrukcja1 Else instrukcja2
```

```
If warunek Then  
  instrukcja1-1  
  ...  
  instrukcja1-N
```

```
Else
```

```
  instrukcja2-1  
  ...  
  instrukcja2-N
```

```
End If
```



Operatory relacyjne i logiczne

Operatory relacyjne

- > większy
- < mniejszy
- >= większy lub równy
- <= mniejszy lub równy
- <> różny
- = równy

Operatory logiczne

- And** koniunkcja
- Or** alternatywa
- Not** negacja

x	y	x And y	x Or y	Not x
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Przykład (x jest zmienną liczbową)

- Sprawdzenie czy x jest liczbą dodatnią: $x > 0$
- Sprawdzenie czy x jest różna od zera: $x <> 0$
- Sprawdzenie czy $x \in [-3, 5]$: $x >= -3$ **And** $x <= 5$
- Sprawdzenie czy $x \in (-\infty, -2] \cup (10, +\infty)$: $x <= -2$ **Or** $x > 10$

Procedura ustawia kolor zielony w komórkach zawierających wartości dodatnie i zero, czerwony w komórkach zawierających wartości ujemne.

```
Public Sub Koloruj1()  
    If ActiveCell.Value >= 0 Then ActiveCell.Font.Color = vbGreen _  
    Else ActiveCell.Font.Color = vbRed  
End Sub
```

Procedura ustawia kolor zielony oraz pogrubienie w komórkach zawierających wartości dodatnie i zero, czerwony oraz kursywę w komórkach zawierających wartości ujemne.

```
Public Sub Koloruj2()  
    If ActiveCell.Value >= 0 Then  
        ActiveCell.Font.Color = vbGreen  
        ActiveCell.Font.Bold = True  
    Else  
        ActiveCell.Font.Color = vbRed  
        ActiveCell.Font.Italic = True  
    End If  
End Sub
```

	A	B	C	D
1				
2		10	a	!
3		-5		
4				

Uwaga: tekst jest interpretowany jako liczba dodatnia.

Kod dostępny na stronie przedmiotu

Wybrane funkcje informacyjne VBA

- `IsDate(w)` – określa czy `w` może być konwertowane na typ `Date`,
- `IsEmpty(w)` – określa czy `w` jest puste/zainicjowane (tylko typ `Variant`),
- `IsNumeric(w)` – określa czy `w` jest liczbą (pusta komórka jest liczbą),
- `isObject(w)` – określa czy `w` reprezentuje obiekt.

*Uwaga: funkcje **Is...** zwracają wartość logiczną **True/False**.*

Określenie nazwy/numeru typu danych

- `TypeName(zmienna)` – zwraca nazwę typu zmiennej jako tekst,
- `VarType(zmienna)` – zwraca kod typu zmiennej.

Kod	Nazwa
0	Empty
1	Null
2	Integer
3	Long
4	Single
5	Double
6	Currency
7	Date

Kod	Nazwa
8	String
9	Object
10	Error
11	Boolean
12	Variant
13	Object
14	Decimal
17	Byte

Modyfikacja procedury `Koloruj2`. Dodatkowy warunek zapobiega modyfikacji ustawień czcionki, gdy wartość w komórce nie jest liczbą (warunek `.Value >= 0` będzie sprawdzany tylko gdy komórka zawiera liczbę).

```
Public Sub Koloruj3()
```

```
    With ActiveCell
```

```
        If Not IsEmpty(.Value) And IsNumeric(.Value) Then
```

```
            If .Value >= 0 Then
```

```
                .Font.Color = vbGreen
```

```
                .Font.Bold = True
```

```
            Else
```

```
                .Font.Color = vbRed
```

```
                .Font.Italic = True
```

```
            End If
```

```
        End If
```

```
    End With
```

```
End Sub
```

	A	B	C	D
1				
2		10		a
3		-5		
4				

Kod dostępny na stronie przedmiotu

Zagnieżdżanie instrukcji sterujących

Zagnieżdżenie instrukcji – umieszczenie pewnej instrukcji sterującej w zasięgu działania innej instrukcji sterującej.

Przykład

```

If warunek1 Then                                     ' pierwsza instr. warunkowa
┌ if warunek2 Then instrukcja1                         ' druga instr. warunkowa
├ Else
│ ┌ If warunek3 Then                                     ' trzecia instr. warunkowa
│ │   instrukcja2
│ │ ┌ Else
│ │ │   instrukcja3
│ │ └ End If
└ End If

```

Druga instrukcja warunkowa zostanie wykonana jeżeli warunek1 jest prawdziwy (wykonanie instrukcji1 zależy od warunku1 i warunku2)

Trzecia instrukcja warunkowa zostanie wykonana jeżeli warunek1 jest fałszywy (wykonanie instrukcji2 i instrukcji3 zależy od warunku1 i warunku3)

Wcięcie – dodatkowy odstęp od lewego marginesu wprowadzany dla zagnieżdżonej instrukcji. Nie wpływa na wykonanie programu, zwiększa czytelność kodu.

Przykład – MsgBox jako funkcja

MsgBox może być wywołany jako funkcja. W takim przypadku zwraca informację o naciśniętym przycisku i może być wykorzystany do komunikacji z użytkownikiem (szczegóły składni s.2-10).

```
MsgBox(prompt, buttons, title) As Integer
```

Dostępne przyciski (argument buttons): vbOKOnly, vbOKCancel, vbYesNoCancel, vbYesNo, vbRetryCancel, vbAbortRetryIgnore.

Zwracane wartości: vbOK, vbCancel, vbAbort, vbRetry, vbIgnore, vbYes, vbNo.

Przykład

Procedura zmienia kolor tekstu w komórce C11 po akceptacji użytkownika

```
Public Sub ZmieńKolor()
```

```
    Dim odp As Integer
```

```
    odp = MsgBox("Zmienić kolor w C11?", vbQuestion + vbYesNo )
```

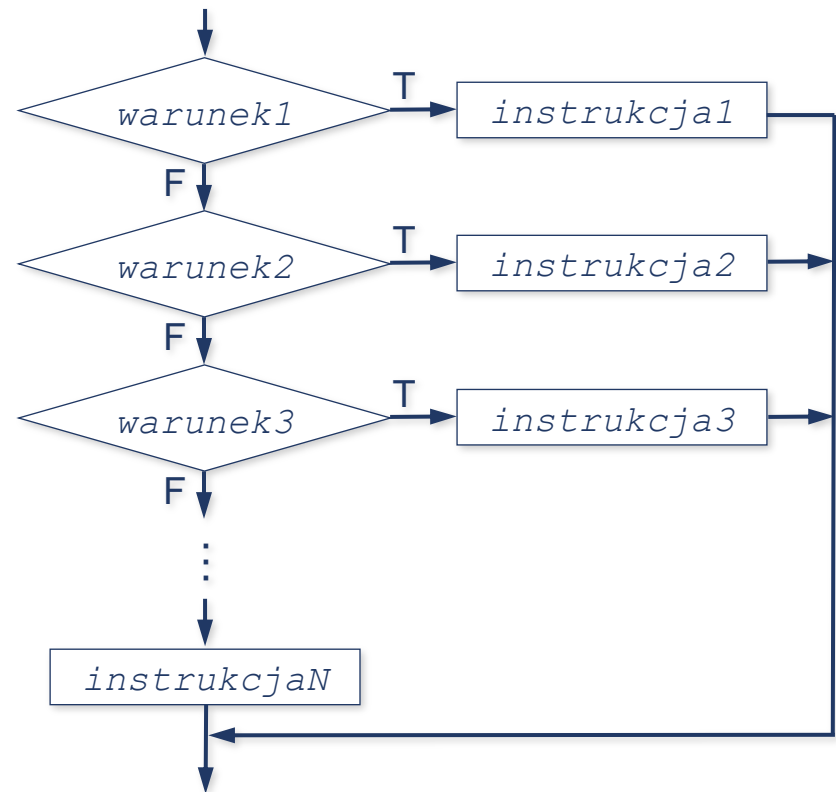
```
    If odp = vbYes Then Range("C11").Font.Color = vbRed
```

```
End Sub
```

Konstrukcja ElseIf

ElseIf jest opcjonalnym składnikiem instrukcji **If-End If**. Może wystąpić wielokrotnie, każdy składnik wprowadza dodatkowy warunek.

```
If warunek1 Then  
  instrukcja1-1  
  ...  
  instrukcja1-N  
ElseIf warunek2 Then  
  instrukcja2-1  
  ...  
  instrukcja2-N  
ElseIf warunek3 Then  
  instrukcja3-1  
  ...  
  instrukcja3-N  
ElseIf ...  
Else  
  instrukcjaN-1  
  ...  
  instrukcjaN-N  
End If
```



Uwaga: warunek $i+1$ będzie sprawdzony jeżeli fałszywy był warunek i -ty. W każdym przebiegu może zostać wykonana co najwyżej jedna grupa instrukcji. Instrukcje w sekcji **Else** (opcjonalne) zostaną wykonane jeżeli wszystkie warunki były fałszywe.

Procedura ustawia kolor w komórkach zawierających liczby zależnie od ich wartości: niebieski dla wartości większych lub równych 100, zielonych dla wartości od 10 do 99, magenta dla wartości od 0 do 10, czerwony dla wartości ujemnych.

```
Public Sub Koloruj3()  
    With ActiveCell  
        If Not IsNumeric(.Value) Then Exit Sub  
        If .Value >= 100 Then .Font.Color = vbBlue  
        If .Value >= 10 And .Value < 100 Then .Font.Color = vbGreen  
        If .Value >= 0 And .Value < 10 Then .Font.Color = vbMagenta  
        If .Value < 0 Then .Font.Color = vbRed  
    End With  
End Sub
```

Kod dostępny na stronie przedmiotu

Uwaga I: Pierwsza instrukcja warunkowa zapobiega przed błędem wykonania (przerywa procedurę gdy komórka zawiera wartość, która nie jest liczbą).

Uwaga II: Kod procedury został zapisany za pomocą czterech instrukcji warunkowych, każdy warunek sprawdzany jest niezależnie, konieczne jest sprawdzenie dolnego i górnego zakresu wartości (kolejność warunków nie jest istotna).

Modyfikacja procedury ze s.3-13 (instrukcja **If-ElseIf**). Kolejne warunki sprawdzane są jeżeli poprzednie były fałszywe, wystarczające jest sprawdzenie dolnego zakresu wartości (kolejność warunków jest istotna).

```
Public Sub Koloruj4()  
    With ActiveCell  
        If Not IsNumeric(.Value) Then Exit Sub  
        If .Value >= 100 Then  
            .Font.Color = vbBlue  
        ElseIf .Value >= 10 Then  
            .Font.Color = vbGreen  
        ElseIf .Value >= 0 Then  
            .Font.Color = vbMagenta  
        Else  
            .Font.Color = vbRed  
        End If  
    End With  
End Sub
```

Kod dostępny na stronie przedmiotu

Przykład – funkcja Wiek

Funkcja określa wiek na podstawie daty urodzenia (niepełnoletni, pełnoletni, emeryt).

```
Public Function Wiek(dataUr As Date) As String
  If dataUr > Date Then
    Wiek = ""
  Else
    If DateAdd("yyyy", 18, dataUr) > Date Then
      Wiek = "niepełnoletni"
    ElseIf DateAdd("yyyy", 65, dataUr) > Date Then
      Wiek = "pełnoletni"
    Else
      Wiek = "emeryt"
    End If
  End If
End Function
```

	B	C	D	E
25				
26		Data urodzenia		
27		30.01.2005		niepełnoletni
28		20.05.1985		pełnoletni
29		10.03.1950		emeryt
30		25.06.2070		
31				

= Wiek(C27)

= Wiek(C28)

= Wiek(C29)

= Wiek(C30)

Date – funkcja zwraca aktualną datę

DateAdd(*interwał*, *liczba*, *data*) – dodaje do podanej *daty* *interwał* czasowy określony *liczbą* (możliwe interwały: yyyy – rok, q – kwartał, m – miesiąc, d – dzień, ww – tydzień, h – godzina, n – minuta, s – sekunda)

Przykład – funkcja Płeć

Funkcja określa płeć na podstawie imienia (działa dla imion polskich).

```
Public Function Płeć(imię As String) As String
    Dim znak As String
    znak = Right(imię, 1)
    If znak = "" Then
        Płeć = ""
    ElseIf znak = "a" Or znak = "A" Then
        Płeć = "kobieta"
    Else
        Płeć = "mężczyzna"
    End If
End Function
```

	G	H	I	J
25				
26		Karol	mężczyzna	= Płeć (H26)
27		Anna	kobieta	= Płeć (H27)
28		Jan	mężczyzna	= Płeć (H28)
29		Monika	kobieta	= Płeć (H29)
30				= Płeć (H29)

Right(tekst, liczba) – funkcja zwraca określoną liczbę znaków z prawej strony tekstu (końcowe znaki tekstu).

Kod dostępny na stronie przedmiotu

Instrukcja wyboru oblicza wartość *wyrażenia* i wykonuje akcję związaną z jednym z przypadków zależnie od wyznaczonej wartości.

Select Case *wyrażenie*

Case *wartość1*

instrukcja1

Case *wartość2 To wartość3*

instrukcja2

Case Is *operator wartość4*

instrukcja3

Case Else

instrukcja4

End Select

Opis przypadków:

- Konkretna wartość (*wartość1*)
- Zakres wartości (*wartość2 To wartość3*)
- Warunek z wykorzystaniem operatora relacyjnego (*Is operator wartość4*)

Jeżeli wartość *wyrażenia* nie odpowiada żadnemu z opisanych przypadków wykonane będą instrukcje umieszczone w sekcji **Case Else** (opcjonalna).

Modyfikacja procedury ze s.14-15 (kolorowanie komórek na podstawie wartości).

```
Public Sub Koloruj5()  
    With ActiveCell  
        If Not IsNumeric(.Value) Then Exit Sub  
        Select Case .Value  
            Case Is >= 100  
                .Font.Color = vbBlue  
            Case 10 To 99  
                .Font.Color = vbGreen  
            Case 0 To 9  
                .Font.Color = vbMagenta  
            Case Else  
                .Font.Color = vbRed  
        End Select  
    End With  
End Sub
```

Kod dostępny na stronie przedmiotu

Funkcje wyboru wartości z listy

Choose(*indeks*, *wartość1*, *wartość2*, ... *wartośćN*)

Funkcja wybiera *wartość* o określonym *indeksie* z listy podanej listy elementów.

Switch(*warunek1*, *wartość1*, *warunek2*, *wartość2*, ... *warunekN*, *wartośćN*)

Argumentami funkcji są pary warunek – wartość. Zwraca wartość odpowiadającą warunkowi o wartości True.

Przykłady

Dim x As Integer, dzień **As String**

```
x = InputBox("Podaj numer dnia tygodnia")
```

```
dzień = Choose(x, "niedziela", "poniedziałek", "wtorek", _  
              "środa", "czwartek", "piątek", "sobota")
```

Dim nr As String, miesiąc **As String**

```
nr = InputBox("Podaj numer miesiąca")
```

```
miesiąc = Switch(nr="I", "styczeń", nr="II", "luty", _  
                nr="III", "marzec", nr="IV", "kwiecień", nr="V", "maj", _  
                nr="VI", "czerwiec", nr="VII", "lipiec", nr="VIII", "sierpień", _  
                nr="IX", "wrzesień", nr="X", "październik", nr="XI", "listopad", _  
                nr="XII", "grudzień")
```

Przykład – funkcja słownie cz.I

```
Public Function Słownie(liczba As Integer) As String

    Dim txt As String
    Dim s As Integer
    Dim d As Integer
    Dim j As Integer

    If Abs(liczba) >= 1000 Then Słownie = "": Exit Function
    If liczba = 0 Then Słownie = "zero": Exit Function

    txt = ""
    If liczba < 0 Then txt = "minus ": liczba = Abs(liczba)

    s = liczba \ 100           'liczba setek
    d = (liczba Mod 100) \ 10 'liczba dziesiątek
    j = liczba Mod 10        'liczba jedności
```

c.d. na następnej stronie

Przykład – funkcja słownie cz.II

```
If s > 0 Then txt = txt & Choose(s, "sto", "dwieście", _  
    "trzysta", "czterysta", "pięćset", "sześćset", _  
    "siedemset", "osiemset", "dziewięćset")  
  
If d > 1 Then txt = txt & " " & Choose(d, "", "dwadzieścia", _  
    "trzydzieści", "czterdzieści", "pięćdziesiąt", _  
    "sześćdziesiąt", "siedemdziesiąt", _  
    "osiemdziesiąt", "dziewięćdziesiąt")  
  
If d = 1 Then txt = txt & " " & Choose(j + 1, "dziesięć", _  
    "jedenaście", "dwanaście", "trzynaście", _  
    "czternaście", "piętnaście", "szesnaście", _  
    "siedemnaście", "osiemnaście", "dziewiętnaście") _  
  
Else txt = txt & " " & Choose(j, "jeden", "dwa", "trzy", _  
    "cztery", "pięć", "sześć", "siedem", "osiem", _  
    "dziewięć")
```

Słownie = txt

End Function

Kod dostępny na stronie przedmiotu