

Techniki programowania



Instrukcje iteracyjne

Przetwarzanie kolekcji – instrukcja For Each
Instrukcje iteracyjne For-Next i Do-Loop

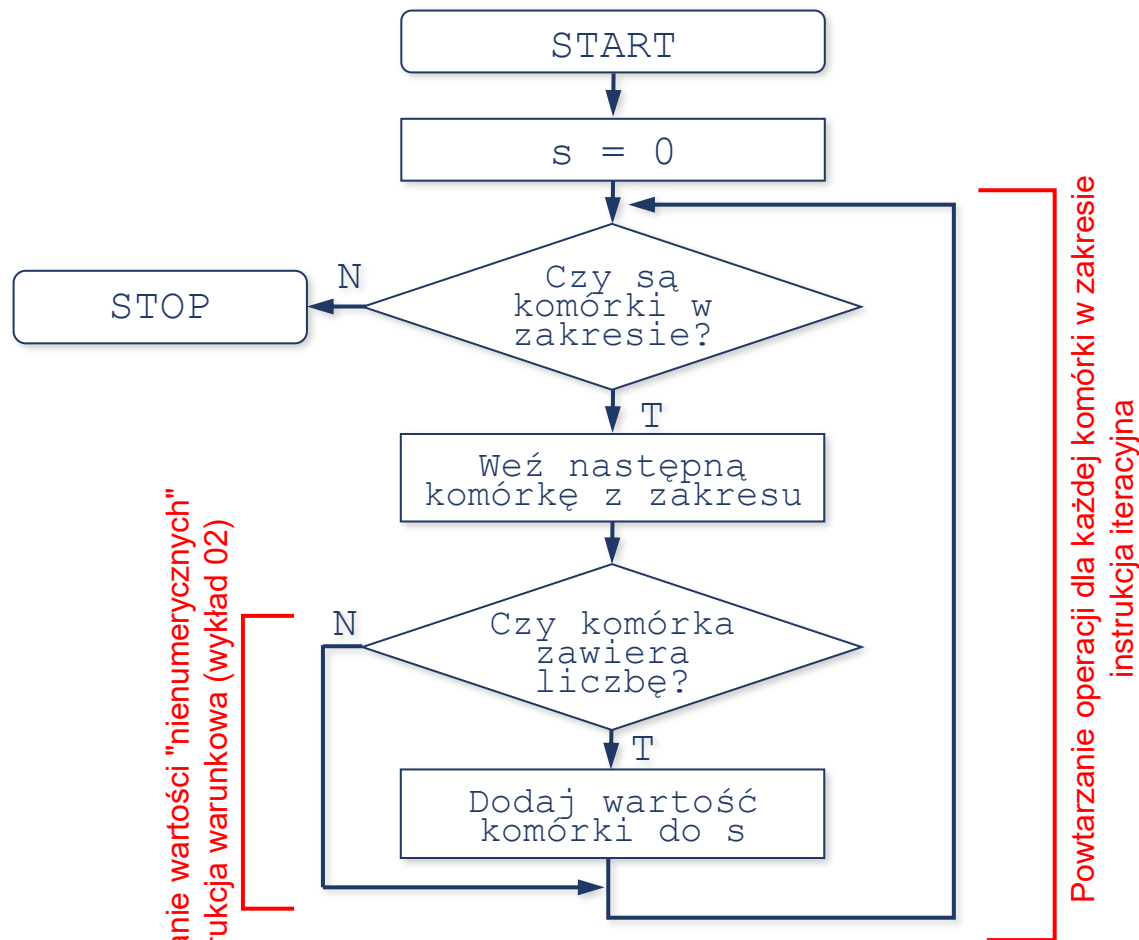
Instrukcje sterujące – SumaLiczb

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)

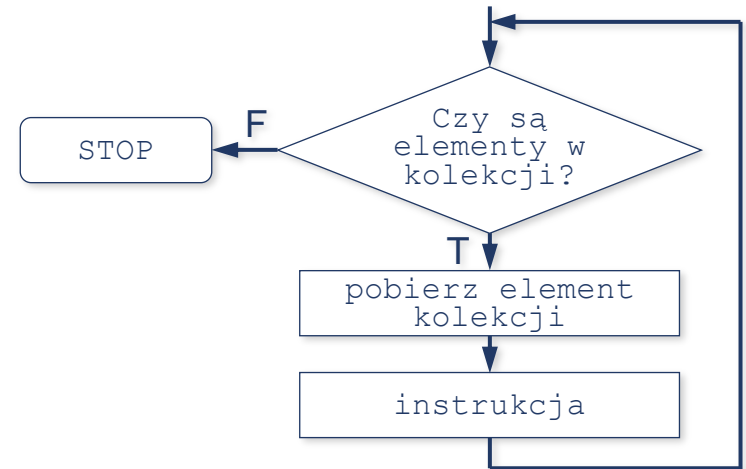
=SumaLiczb(C1:C7)



Przetwarzanie elementów kolekcji – For Each

Wykonaj instrukcję (grupę instrukcji) dla wszystkich elementów kolekcji.

```
For Each zmienna In kolekcja  
    instrukcja1  
    instrukcja2  
    ...  
    instrukcjaN  
Next
```



Pętla powtarza ciąg instrukcji tyle razy ile jest elementów w *kolekcji*. W kolejnych iteracjach (powtórzeniach) *zmienna* reprezentuje kolejny element *kolekcji*.

Wybrane kolekcje VBA

- `Range`, `Cells` – kolekcja komórek arkusza,
- `Selection` – kolekcja komórek w aktualnie wybranym zakresie,
- `Rows`, `Columns` – kolekcja wierszy/kolumn,
- `Workbooks` – kolekcja aktualnie otwartych dokumentów,
- `Worksheets` – kolekcja arkuszy wybranego dokumentu.

Przykład – kolorowanie komórek

Procedura modyfikuje kolor wszystkich komórek we wskazanym zakresie, które zawierają wartości liczbowe (powtarza operacje zapisane w procedurze Koloruj3, w.02, s.25 dla wszystkich komórek w zakresie).

```
Public Sub KolorujKomórki1()  
    Dim z As Range  
    Dim k As Range  
    On Error GoTo Wycofanie wybór zakresu  
    Set z = Application.InputBox("Wskaż zakres", Type:=8)  
    For Each k In z  
        If Not IsEmpty(k.Value) And IsNumeric(k.Value) Then  
            If k.Value >= 0 Then k.Font.Color = vbGreen _  
            Else k.Font.Color = vbRed  
            End If ustawienie koloru gdy komórka zawiera liczbę  
        Next  
    Exit Sub  
Wycofanie:  
End Sub
```

pętla

Kod dostępny na stronie przedmiotu

Przykład – analiza

```
For Each k In z
  If Not IsEmpty(k.Value) And IsNumeric(k.Value) Then
    If k.Value >= 0 Then
      k.Font.Color = vbGreen
    Else
      k.Font.Color = vbRed
    End If
  End If
Next
```

Range(B2:B7)
kolekcja komórek

	A	B	C
1			
2		5	
3			
4		-4	
5		abc	
6		9	
7		01.01.2000	
8			

Integer
Empty
Integer
String
Integer
Date

Analiza (z=Range('B2:B7'))

1. k = Range('B2')
wartość niepusta, liczba > 0
Color = vbGreen
2. k = Range('B3')
wartość pusta
kolor nie jest modyfikowany
3. k = Range('B4')
wartość niepusta, liczba < 0
Color = vbRed
4. k = Range('B5')
wartość niepusta, string
kolor nie jest modyfikowany
5. k = Range('B6')
wartość niepusta, liczba > 0
Color = vbGreen
6. k = Range('B7')
wartość niepusta, data
kolor nie jest modyfikowany

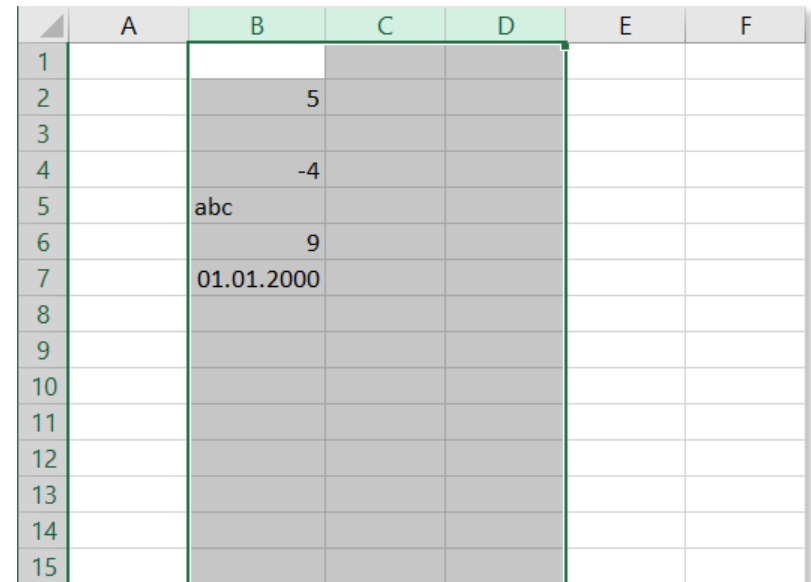
Przetwarzanie dużych zakresów komórek

Wybór dużego zakresu komórek może prowadzić do istotnego wydłużenia czasu wykonania makra (twórca makra nie ma wpływu na wielkość zakresu wybranego przez użytkownika). W wielu przypadkach zaznaczony zakres zawiera komórki puste lub zawierające wartości, które nie są przetwarzane, jednak ich obecność wymusza dodatkowe (puste) iteracje pętli.

Przykład

Zaznaczenie obejmuje kolumny od B do D, każda kolumna zawiera 1 048 576 komórek (Excel 2019). Wykonanie procedury `KolorujKomórki1` wymaga:

- $3 \times 1\,048\,576 = 3\,145\,728$ iteracji,
- czas wykonania (i7 3.4GHz) ok. 25s
- komórki zawierające wartości: 5
- w tym wartości liczbowe: 3
- 3 145 725 iteracji pustych



	A	B	C	D	E	F
1						
2		5				
3						
4		-4				
5		abc				
6		9				
7		01.01.2000				
8						
9						
10						
11						
12						
13						
14						
15						

Działania na zakresach komórek

Wybór komórek arkusza z wartościami określonego typu (metoda obiektu Range)

```
SpecialCells(Type As XlCellType, _  
             Value As XlSpecialCellsValue) As Range
```

Type określa typ wartości:

- `xlCellTypeBlanks` – puste komórki
- `xlCellTypeConstants` – komórki zawierające wartości stałe
- `xlCellTypeFormulas` – komórki zawierające formuły
- `xlCellTypeVisible` – komórki widoczne

Value (opcjonalny) określa szczegółowy typ wartości:

- `xlErrors` – komórki z błędami
- `xlLogical` – komórki z wartościami logicznymi
- `xlNumbers` – komórki z wartościami liczbowymi
- `xlTextValues` – komórki z wartościami tekstowymi

Operacje na zakresach (metody obiektu Application)

```
Union(z1 As Range, z2 As Range, z3, z4, ... z30)
```

```
Intersect(z1 As Range, z2 As Range, z3, z4, ... z30)
```

Metody wyznaczają sumę oraz część wspólną zakresów `z1...z30` przekazanych jako argumenty (wymagane co najmniej dwa niepuste zakresy).

Przykład – kolorowanie komórek, wersja 2

Modyfikacja procedury `KolorujKomórki1` (s.4). Wskazany przez użytkownika zakres jest ograniczony tylko do komórek zawierających wartości numeryczne.

```
Public Sub KolorujKomórki2()  
    Dim z As Range  
    Dim k As Range  
    On Error GoTo Wycofanie  
    Set z = Application.InputBox("Wskaż zakres", Type:=8)  
    Set z = z.SpecialCells(xlCellTypeConstants, xlNumbers)  
    For Each k In z  
        If IsNumeric(k.Value) Then  
            If k.Value >= 0 Then k.Font.Color = vbGreen _  
            Else k.Font.Color = vbRed  
        End If  
    Next  
    Exit Sub  
Wycofanie:  
End Sub
```

wskazanie zakresu i wybór komórek, które zawierają wartości numeryczne

Uwaga: sprawdzenie czy komórka jest niepusta w tym przypadku nie jest potrzebne, ze względu na użycie metody `SpecialCells` zakres zawiera wyłącznie wartości numeryczne (w tym daty).

Kod dostępny na stronie przedmiotu

Funkcja SumaLiczb

Funkcja oblicza sumę wartości liczbowych (z wyłączeniem dat) w podanym zakresie.

```
Public Function SumaLiczb1(zakres As Range) As Double  
    Dim k As Range  
    Dim s As Double  
    s = 0  
    For Each k In zakres  
        If Not IsEmpty(k.Value) And IsNumeric(k.Value) Then  
            s = s + k.Value  
        End If  
    Next  
    SumaLiczb1 = s  
End Function
```

Uwaga: funkcja może być uzupełniona o wstępny wybór komórek zawierających wartości numeryczne. Obydwie wersje są dostępne na stronie przedmiotu.

	J	K	L
26			
27		SumaLiczb1	
28		1	
29		2	
30	a		
31		1 mar 22	
32	b		
33		3	
34		4	
35		10	

=SumaLiczb1(K28:K34)

Analiza, SumaLiczb1("K28:K34")

0. s=0
1. k=Range("K28"), s=0+1=1
2. k=Range("K29"), s=1+2=3
3. k=Range("K30")
4. k=Range("K31")
5. k=Range("K32")
6. k=Range("K33"), s=3+3=6
7. k=Range("K34"), s=6+4=10

Kod dostępny na stronie przedmiotu

Przykład – kolorowanie wierszy

Procedura ustawia kolor tła w parzystych wierszach wskazanego zakresu.

```
Public Sub KolorujWiersze1()  
    Dim z As Range  
    Dim w As Range  
    Dim i As Integer  
  
    On Error GoTo Wycofanie  
    Set z = Application.InputBox(...)
```

```
i = 1
```

```
For Each w In z.Rows
```

```
    If i Mod 2 = 0 Then w.Interior.Color = RGB(255,255,225) _
```

```
    Else w.Interior.ColorIndex = xlColorIndexNone
```

```
    i = i + 1
```

```
Next
```

Wycofanie:

```
End Sub
```

Uwaga 1: pętla wykonuje operacje dla każdego wiersza w wybranym zakresie (kolekcja `z.Rows`).

Uwaga 2: zmienna `i` jest powiększana w każdym przebiegu pętli (jest licznikiem wierszy).

`i Mod 2 = 0` gdy numer wiersza jest parzysty.

	J	K	L	M	N	O	P	Q
5								
6			A	B	C	D	E	
7		1	125	3	226	-23	-7	
8		2	3	4	37	-5	345	
9		3	-7	23	23	45	3	
10		4	0	-11	12	120	45	
11		5	22	0	-2	38	22	
12		6	24	22	3	56	2	
13		7	23	34	234	2	3	
14								

Kod dostępny na stronie przedmiotu

Przykład – przetwarzanie kolekcji arkuszy

Procedura wypełnia pierwszą komórkę każdego arkusza w bieżącym dokumencie wartością zawierającą numer arkusza i jego nazwę.

```
Public Sub NazwyArkuszy ()
```

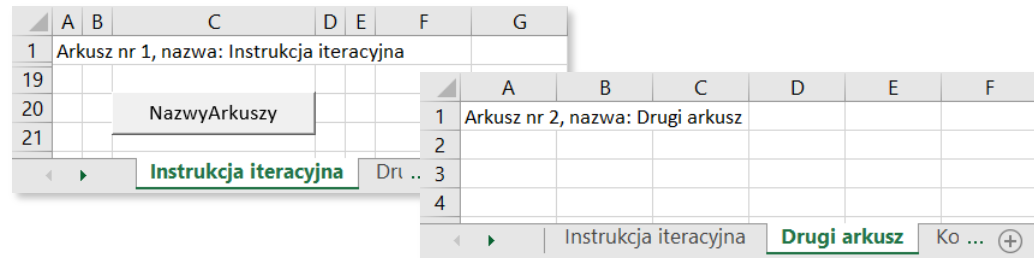
```
    Dim arkusz As Worksheet
```

```
    For Each arkusz In ActiveWorkbook.Worksheets
```

```
        arkusz.Cells(1, 1).Value = "Arkusz nr " & arkusz.Index & _  
            ", nazwa: " & arkusz.Name
```

```
    Next
```

```
End Sub
```



Uwaga 1: pętla wykonuje operacje dla każdego arkusza w bieżącym dokumencie (kolekcja `ActiveWorkbook.Worksheets`).

Uwaga 2: Operator „&” łączy wartości tekstowe tworząc pojedynczy łańcuch znakowy (konkatenacja łańcuchów), np. "abc" & "def" = "abcdef".

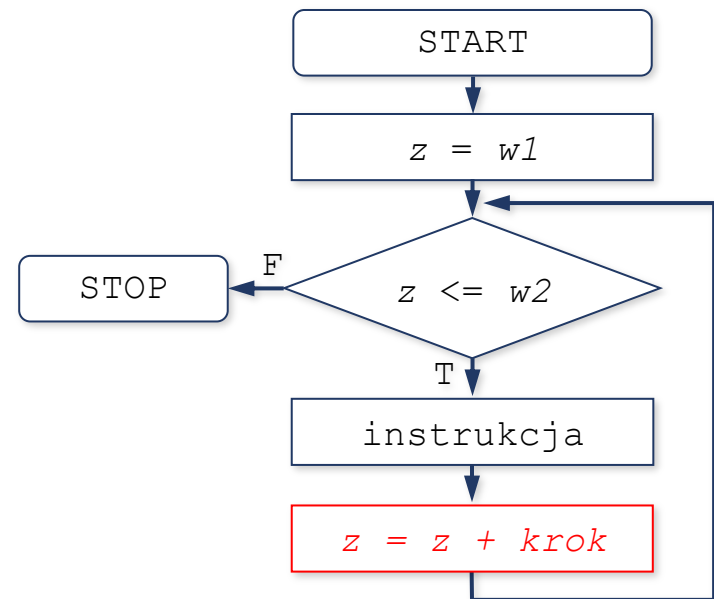
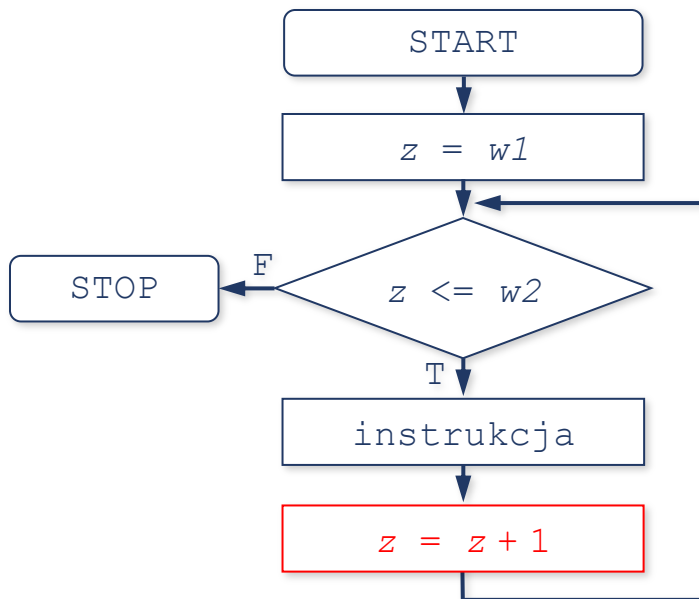
Kod dostępny na stronie przedmiotu

Instrukcja iteracyjna For-Next

```
For zmienna = w1 To w2  
  instrukcja1  
  ...  
  instrukcjaN  
Next zmienna
```

```
For zmienna = w1 To w2 Step krok  
  instrukcja1  
  ...  
  instrukcjaN  
Next zmienna
```

Dla *zmiennnej* przyjmującej wartości od *w1* do *w2* (z domyślnym krokiem 1 lub określonym przez *krok*) wykonuj ciąg instrukcji.



Przykład – kolorowanie wierszy wersja II

Modyfikacja procedury KolorujWiersze1 (s.10) – instrukcja **For-Next**

```
Public Sub KolorujWiersze2()  
    Dim z As Range  
    Dim i As Integer  
  
    On Error GoTo Wycofanie  
    Set z = Application.InputBox(...)  
    On Error GoTo 0  
  
    For i = 1 To z.Rows.Count  
        If i Mod 2 = 0 Then  
            z.Rows(i).Interior.Color = RGB(255, 255, 225)  
        Else  
            z.Rows(i).Interior.ColorIndex = xlColorIndexNone  
        End If  
    Next i  
  
    Wycofanie:  
End Sub
```

	F	G	H	I	J	K	L
10							
11		A	B	C	D	E	
12	1	125	3	226	-23	-7	
13	2	3	4	37	-5	345	
14	3	-7	abc	23	pqr	3	
15	4	0	-11	xyz	120	45	
16	5	22	0	-2	38	abcxyz	
17	6	24	22	3	56	2	
18	7	167	18	287	186	388	
19							

Przykład – kolorowanie wierszy wersja III

Modyfikacja procedury KolorujWiersze1 (s.10) – instrukcja **For-Step-Next**

```
Public Sub KolorujWiersze3()  
    Dim z As Range  
    Dim i As Integer  
  
    On Error GoTo Wycofanie  
    Set z = Application.InputBox(...)  
    On Error GoTo 0
```

	F	G	H	I	J	K	L
10							
11		A	B	C	D	E	
12	1	125	3	226	-23	-7	
13	2	3	4	37	-5	345	
14	3	-7	abc	23	pqr	3	
15	4	0	-11	xyz	120	45	
16	5	22	0	-2	38	abcxyz	
17	6	24	22	3	56	2	
18	7	167	18	287	186	388	
19							

```
z.Interior.ColorIndex = xlColorIndexNone
```

```
For i = 2 To z.Rows.Count Step 2
```

```
    z.Rows(i).Interior.Color = RGB(255, 255, 225)
```

```
Next i
```

Wycofanie:

```
End Sub
```

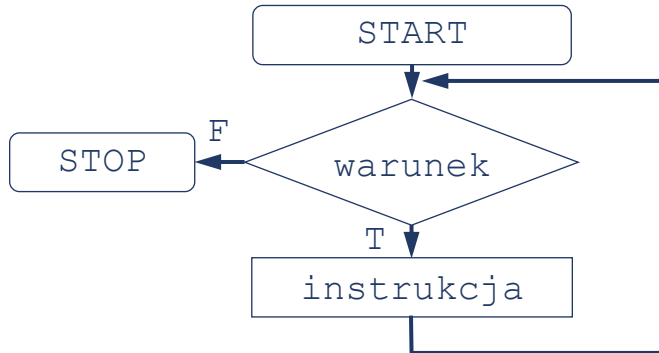
Uwaga: Pętla wykonuje instrukcję tylko dla parzystych wierszy (zaczyna od drugiego i powiększa zmienną o 2). W tym przypadku sprawdzanie parzystości numeru wiersza nie jest konieczne.

Kod dostępny na stronie przedmiotu

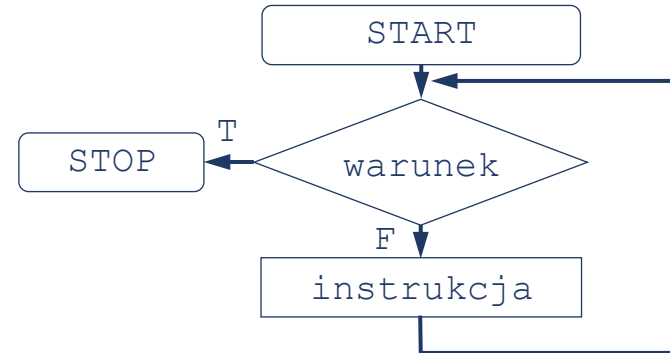
Warianty składni instrukcji Do-Loop

Do While <i>warunek</i> instrukcja Loop	Instrukcja jest wykonywana dopóki <i>warunek</i> jest spełniony (przerywa gdy ma wartość <code>False</code> , sprawdzenie na początku)
Do Until <i>warunek</i> instrukcja Loop	Instrukcja jest wykonywana chyba że <i>warunek</i> jest spełniony (przerywa gdy ma wartość <code>True</code> , sprawdzenie na początku)
Do Loop instrukcja While <i>warunek</i>	Instrukcja jest wykonywana dopóki <i>warunek</i> jest spełniony (przerywa gdy ma wartość <code>False</code> , sprawdzenie na końcu)
Do Loop instrukcja Until <i>warunek</i>	Instrukcja jest wykonywana chyba że <i>warunek</i> jest spełniony (przerywa gdy ma wartość <code>True</code> , sprawdzenie na końcu)

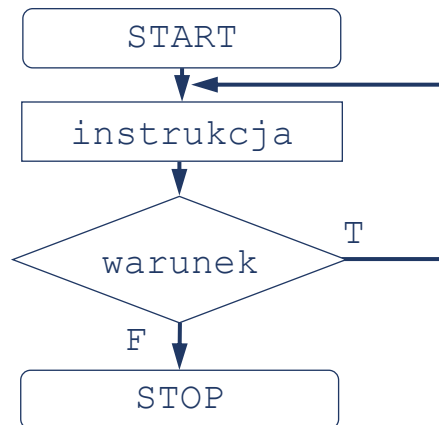
Instrukcje iteracyjne Do-Loop



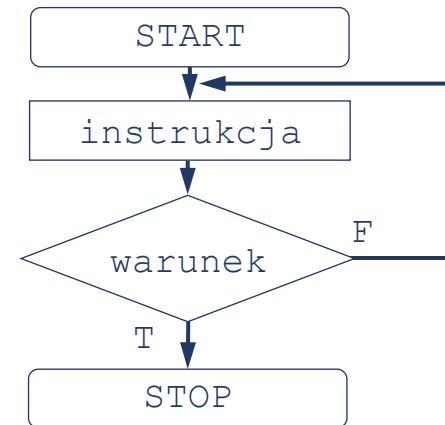
Do While - Loop



Do Until - Loop



Do Loop - While



Do Loop - Until

Przykład – poszukiwanie wartości, wersja I

Procedura poszukuje pierwszego wystąpienia liczby w bieżącym wierszu przeszukując komórki położone na prawo od komórki aktualnej, pętla Do-Loop-Until.

```
Public Sub ZnajdźLiczbę1()
```

```
    Dim k As Range
```

```
    Set k = ActiveCell
```

```
    On Error GoTo koniec
```

```
    Do
```

```
        Set k = k.Offset(0, 1)
```

```
    Loop Until Not IsEmpty(k.Value) And IsNumeric(k.Value)
```

```
    k.Select
```

```
koniec:
```

```
End Sub
```

	B	C	D	E	F	G	H	I	J
33									
34			ZnajdźLiczbę		a	b	2	3	d
35									
36									

Analiza wykonania programu

1. k = "F34"

2. Wykonanie pętli:

2.1. k = k.Offset(0,1) = "G34"

2.2. Not IsEmpty = True, IsNumeric = False

2.3. k = k.Offset(0,1) = "H34"

2.4. Not IsEmpty = True, IsNumeric = True

3. k.Select

Kod dostępny na stronie przedmiotu

Przykład – poszukiwanie wartości, wersja II

Procedura poszukuje pierwszego wystąpienia liczby w bieżącym wierszu przeszukując komórki położone na prawo od komórki aktualnej, pętla Do-Loop-While.

```
Public Sub ZnajdźLiczbę2()  
    Dim k As Range  
    Set k = ActiveCell  
    On Error GoTo koniec  
    Do  
        Set k = k.Offset(0, 1)  
    Loop While IsEmpty(k.Value) Or Not IsNumeric(k.Value)  
    k.Select  
koniec:  
End Sub
```

Uwaga: Pętla Do-Loop-Until przerywa działanie gdy warunek jest prawdziwy, Do-Loop-While gdy warunek jest falszywy. Dowolną operację zapisaną przy pomocy jednej pętli można zamienić na drugą przez zanegowanie warunku.

Kod dostępny na stronie przedmiotu

Funkcja Słownie – używane elementy VBA

Operatory arytmetyczne

+	dodawanie	\	dzielenie całkowite
-	odejmowanie	mod	dzielenie modulo
*	mnożenie	^	potęgowanie
/	dzielenie	()	grupowanie działań

Operator konkatencji (łączenia)

& łączy wartości typu String wykonując konwersję jeżeli jest konieczna

Funkcje wyboru wartości z listy

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

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

`Switch(warunek1, wartość1, warunek2, wartość2, ... war.N, wart.N)`

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

Przykład – funkcja Słownie cz.I

Funkcja prywatna zamienia na tekst liczby od 0 do 999. Używana przez właściwą funkcję Słownie do zamiany kolejnych trójek (tysięcy, milionów, itd.)

```
Private Function Słownie_pomoc(liczba As Integer) As String
    Dim txt As String
    Dim s As Integer, d As Integer, j As Integer
    s = liczba \ 100           'liczba setek
    d = (liczba Mod 100) \ 10 'liczba dziesiątek
    j = liczba Mod 10        'liczba jednośc
    If s > 0 Then txt = txt & Choose(s, "sto", "dwieście",...)
    If d > 1 Then txt = txt & " " & Choose(d, "", "dwadzieścia",...)
    If d = 1 Then
        txt = txt & " " & Choose(j + 1, "dziesięć", "jedenaście",...)
    Else
        txt = txt & " " & Choose(j, "jeden", "dwa", "trzy",...)
    End If
    Słownie_pomoc = txt
End Function
```

Przykład – funkcja Słownie cz.II

```
Public Function Słownie(liczba As Long) As String
    Dim txt As String
    Dim l As Integer, i As Integer
    Dim ujemna As Boolean
    If liczba = 0 Then: Słownie = "zero": Exit Function
    If liczba < 0 Then: liczba = Abs(liczba): ujemna = True
    txt = ""
    i = 1
    Do While liczba > 0
        l = liczba Mod 1000
        liczba = liczba \ 1000
        txt = Słownie_pomoc(l) & _
            Choose(i, "", " tys.", " mln", " mld") & " " & txt
        i = i + 1
    Loop
    If ujemna Then txt = "minus " & txt
    Słownie = txt
End Function
```

Kod dostępny na stronie przedmiotu