



Pętle

Program wykładu

- Pojęcie pętli
- Pętla **FOR**
- Pętla **DO...LOOP**
- Pętle zagnieżdżone

Pojęcie pętli

- Suma lub iloczyn dowolnych n liczb wprowadzanych z klawiatury.
- Wprowadzanie liczb z klawiatury; po wprowadzeniu liczby 0 program powinien podać sumę wprowadzonych liczb.
- Suma liczb większych od zadanej spośród n wprowadzonych.
- Ilość liczb większych od zadanej spośród n wprowadzonych.
- Znajdowanie największego elementu w wektorze.
- Wybór najmniejszego elementu w macierzy wraz z wyznaczeniem indeksów tego elementu.

Rodzaje pętli

- Instrukcje iteracyjne są określone wspólnym mianem pętli. Istnieją trzy odmiany pętli:
 1. liczona,
 2. z logicznym warunkiem początku/zakończenia,
 3. typu obiektowego.
- Pętla liczona jest konstrukcją powtarzającą wykonanie pojedynczego bloku kodu określoną ilość razy. Pętle liczone są wykorzystywane wówczas, gdy **z góry wiadomo, ile razy ma się wykonać jakiś fragment programu.**

Pętla liczona

For *zmienna_ster = początek* **To** *koniec* [**Step** *krok*]

·
Blok instrukcji

·
[**Exit For**]

·
Blok instrukcji

·
Next *zmienna_ster*

- W *bloku instrukcji* nie należy zmieniać wartości zmiennej sterującej.

Pętla liczona

```
For zm_ster = pocz To kon      zm_ster = pocz
.                                1: If zm_ster > kon Then GoTo 10
.                                .
.                                Blok instrukcji
.                                .
.                                [Exit For]
.                                [????]
.                                .
.                                Blok instrukcji
.                                .
Next zm_ster                    zm_ster = zm_ster + 1
.                                GoTo 1
.                                10: ` następna instrukcja po Next
```

Pętla liczona

$x = 3$

For $i = 1$ **To** 7 **Step** 2

$x = x + i$

If $x > 7$ **Then Exit For**

Next i

- Jaka będzie wartość zmiennych x oraz i po wykonaniu tej pętli?

Pętle warunkowe

- Istnieją dwa rodzaje instrukcji z logicznym warunkiem zakończenia **Do...Loop** i **While...Wend**.
- Pętle z logicznym warunkiem zakończenia to pętle, których zakończenie następuje dopiero w momencie spełnienia pewnego warunku. Podczas gdy pętle liczone wykonują jakiś blok kodu konkretną liczbę razy, pętle z logicznym warunkiem zakończenia mogą się wykonywać nawet nieskończoną liczbę razy albo nie wykonywać się w ogóle w zależności od stanu warunku logicznego kończącego taką pętlę.
- Podstawową pętlą z logicznym warunkiem zakończenia jest pętla **Do...Loop**. Istnieją cztery warianty tej pętli:
 1. z warunkiem **True** na początku,
 2. z warunkiem **True** na końcu,
 3. z warunkiem **False** na początku,
 4. z warunkiem **False** na końcu.

Pętle warunkowe

Warunek **True** na początku

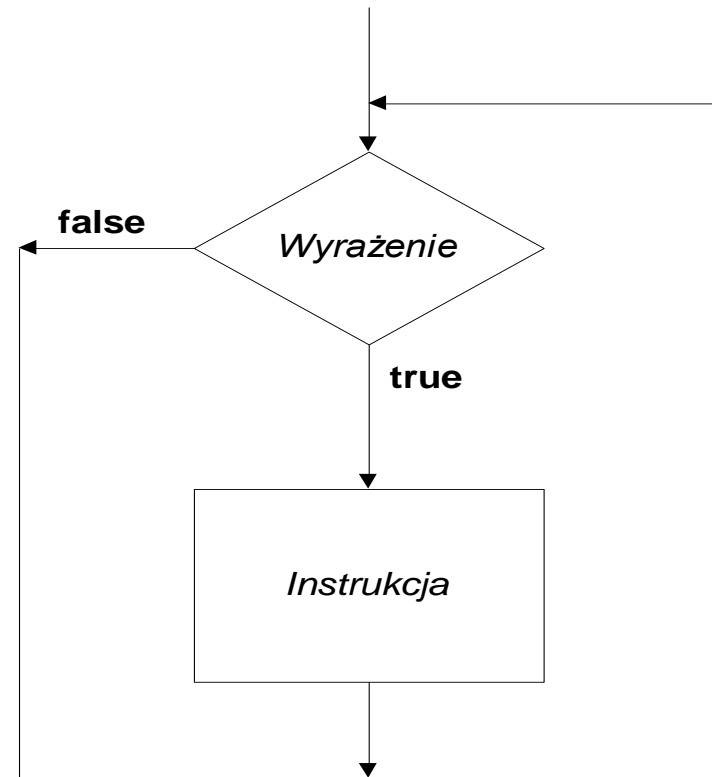
Do While *warunek*

blok instrukcji

[**Exit Do**]

blok instrukcji

Loop



Pętle warunkowe

Warunek **True** na końcu

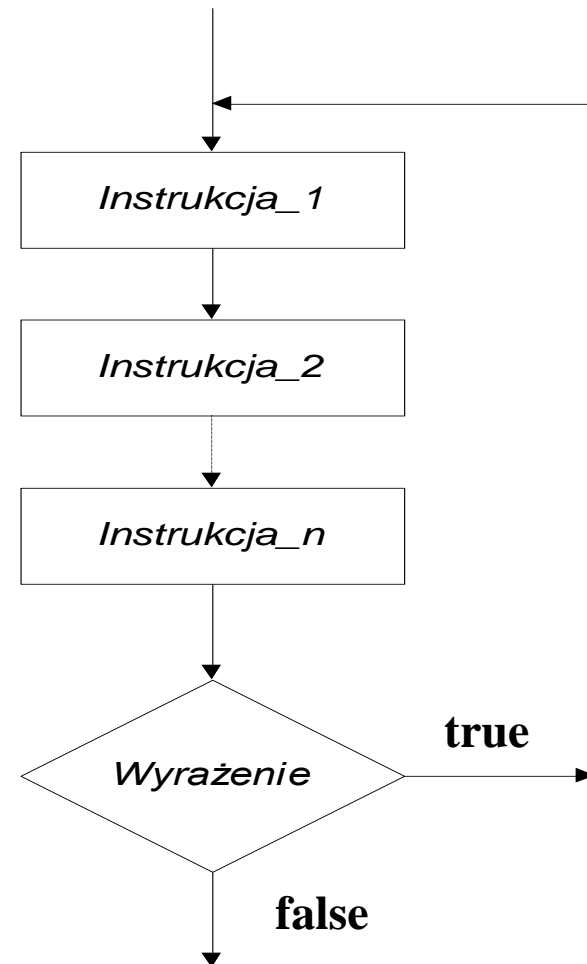
Do

blok instrukcji

[Exit Do]

blok instrukcji

Loop While *warunek*



Pętle warunkowe

Warunek **False** na początku

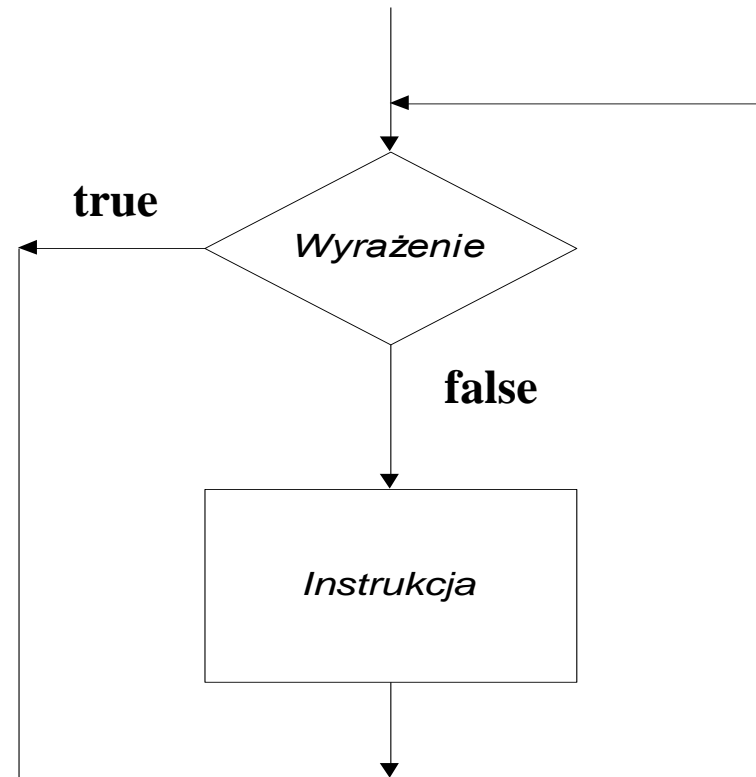
Do Until warunek

blok instrukcji

[Exit Do]

blok instrukcji

Loop



Pętle warunkowe

Warunek **False** na końcu

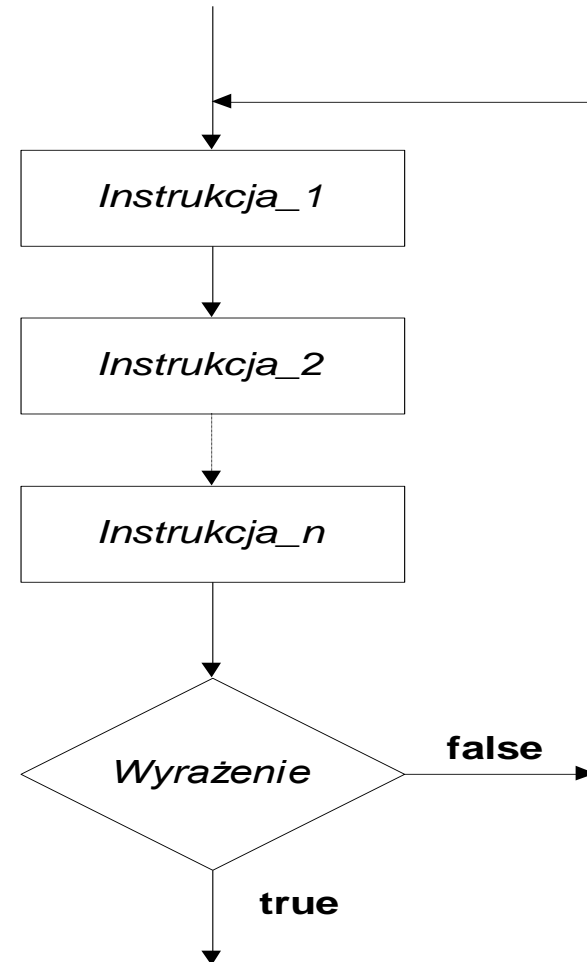
Do

blok instrukcji

[Exit Do]

blok instrukcji

Loop Until *warunek*



Pętle warunkowe

- Słowa kluczowe **While** i **Until** decydują o znaczeniu warunku wykonywania pętli. **While** wskazuje, że pętla będzie się wykonywać, **dopóki warunek** ma wartość **True**, a słowo kluczowe **Until** wskazuje, że pętla będzie się wykonywać, **aż warunek** przyjmie wartość **True**.
- Instrukcja **Exit Do** powoduje wcześniejsze zakończenie pętli i jest zazwyczaj elementem struktury logicznej (instrukcji **If**) sprawdzającej alternatywny warunek zakończenia pętli, taki jak np. wystąpienie błędu.

Pętle warunkowe

Pętla **While - Wend** jest reliktem starszych wersji BASICa. Składnia pętli jest następująca:

While warunek

blok instrukcji

Wend

W konstrukcji tej nie ma instrukcji **EXIT** umożliwiającej wcześniejsze wychodzenie z pętli, ponadto nie można umieszczać warunku na końcu pętli.

Pętle zagnieżdżone

- Elementem bloku instrukcji w pętli może być inna pętla (wewnętrzna, zagnieżdżona) np.:

```
For i = 1 to 3
  For k=1 to 5
    msgbox str(i*k)
  Next k
Next i
```

Wartości zmiennych k oraz i ?

- Pętla wewnętrzna musi być zamknięta przed zamknięciem pętli zewnętrznej.
- Konstrukcja taka jest szczególnie przydatna przy operowaniu na wielowymiarowych tablicach.

Pętle - przykład

- Wyzerować elementy do połowy wektora

0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

0	0
1	0
2	0
3	0
4	0
5	4
6	3
7	2
8	1
9	0

```
Sub zer_wektora_1()  
Dim wekt(10) As Single  
Dim n As Byte  
Dim i As Byte  
n = 9  
For i = 0 To n  
    wekt(i) = n - i  
Next i  
For i = 0 To (n \ 2)  
    wekt(i) = 0  
Next i  
End Sub
```

```
Sub zer_wektora_2()  
Dim wekt(10) As Single  
Dim n As Byte  
Dim i As Byte  
n = 9  
For i = 0 To n  
    wekt(i) = n - i  
Next i  
i = 0  
Do While i <= n / 2 - 1  
    wekt(i) = 0  
    i = i + 1  
Loop  
End Sub
```


Pętle - przykład

- Wyzerować elementy macierzy poniżej i na przekątnej

	0	1	2	3	4
0	1	2	3	4	5
1	2	3	4	5	6
2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9

	0	1	2	3	4
0	0	2	3	4	5
1	0	0	4	5	6
2	0	0	0	6	7
3	0	0	0	0	8
4	0	0	0	0	0

```
Sub zer_tablicy_1()  
Dim tabl(5, 5) As Single  
Dim n As Byte  
Dim i As Byte  
Dim j As Byte  
n = 4  
For i = 0 To n  
    For j = 0 To n  
        tabl(i, j) = i + j + 1  
    Next j  
Next i  
For i = 0 To n  
    For j = 0 To i  
        tabl(i, j) = 0  
    Next j  
Next i  
For i = 0 To n  
    For j = 0 To n  
        Cells(i + 1, j + 1) = tabl(i, j)  
    Next j  
Next i  
End Sub
```

```
Sub zer_tablicy_2()  
Dim tabl(5, 5) As Single  
Dim n As Byte  
Dim i As Byte  
Dim j As Byte  
n = 4  
For i = 0 To n  
    For j = 0 To n  
        tabl(i, j) = i + j + 1  
    Next j  
Next i  
i = 0  
Do While i <= n  
    j = 0  
    Do While j <= i  
        tabl(i, j) = 0  
        j = j + 1  
    Loop  
    i = i + 1  
Loop  
For i = 0 To n  
    For j = 0 To n  
        Cells(i + 1, j + 1) = tabl(i, j)  
    Next j  
Next i  
End Sub
```

Pętle - przykład

- Wprowadzanie danych z kontrolą poprawności

```
Sub opornik()  
Dim r1 As Single  
Dim r2 As Single  
Dim rsz As Single  
Dim rr As Single  
Do  
    r1 = InputBox("Podaj r1 > 0")  
Loop Until r1 > 0  
Do  
    r2 = InputBox("Podaj r2 > 0")  
Loop While r2 <= 0  
rsz = r1 + r2  
rr = r1 * r2 / rsz  
MsgBox "Oporność szeregowo = " & rsz  
MsgBox "Oporność równoległe = " & rr  
End Sub
```

Pętle - przykład

- I „żywe” przykłady:
 1. zsumować n (powiedzmy pierwszych 10) liczb z pierwszej kolumny arkusza
 2. zsumować liczby z pierwszej kolumny arkusza aż do komórki, w której wartość liczby = -1
 3. zsumować liczby z pierwszej kolumny arkusza do momentu, gdy wartość ich sumy osiągnie zadaną wartość np. 1500