

Pętle w Paskalu

Pętla WHILE

1

Pętla - definicja

- **Pętla** to struktura języka programowania, która pozwala na cykliczne wykonanie wiele razy tej samej (lub podobnej) operacji.
- Pętla operuje na takich samych operacjach, danych, strukturach, obiektach czy procedurach.

2

Potrzeba stosowania pętli WHILE

- Pętla FOR służy do wykonania jakiegoś polecenia pewną ilość razy. Czasem jednak potrzeba nam, by pętla wykonywała się ze **względu na jakiś warunek**, a nie pewną ilość razy.

- Dlatego należy zastosować pętlę WHILE. Wykonuje ona jakieś polecenia **dopóki warunek jest spełniony**.

- Ma ona postać:

```
WHILE (Warunek) do  
    Operacja
```

- Przykład

```
While (i < 1000) do  
    AX:=BX;
```

-

3

Pętla WHILE

- WHILE - DO jest instrukcją pętli z kontrolowanym WEJŚCIEM. Jej działanie polega na obliczeniu wartości warunku i dopiero wtedy wykonaniu instrukcji lub ciągu instrukcji.
- Warunek jest sprawdzany **na początku** wykonania serii instrukcji występujących po DO i zawierających się pomiędzy **BEGIN** i **END**; - zwróćmy uwagę że END jest zakończony średnikiem.
- Warunek zwraca wartość logiczną, to znaczy
 - Jest spełniony - prawdę
 - Nie jest spełniony - fałsz.
- Seria instrukcji będzie wykonywana dopóki warunek jest spełniony.
- Jak widać, to jak długo będą wykonywane instrukcje jest uzależnione od sterowania warunkiem pętli w tych że instrukcjach.

4

Składnia pętli WHILE

WHILE (warunek) DOPÓKI warunek jest spełniony

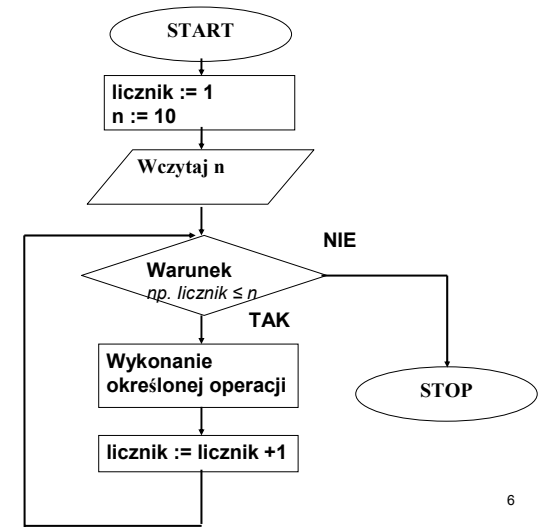
DO
BEGIN

instrukcja 1;
instrukcja 2;
....
instrukcja n;
END;

WYKONUJ

instrukcja 1;
instrukcja 2;
....
instrukcja n;
KONIEC

5



6

Ogólna zasada działania pętli

PĘTLA WHILE

1. **Start** - uruchomienie programu
2. inicjalizacja wstępna zmiennych - *licznik:=1, n:=10*
3. **wczytaj n** - program wczytuje n
4. *licznik ≤ n* - **postawienie warunku**
 - a) **Nie** - niespełnienie warunku ($\text{licznik} \leq n$) zatrzymanie programu
 - b) **Tak** - spełnienie warunku ($\text{licznik} > n$) dalsza praca programu
5. Wykonanie przewidzianej operacji.
6. Operacja zwiększania licznika o 1 (INKREMENTACJA)
7. Powrót do sprawdzenia warunku, dwie możliwości
 - a) **NIE** - zakończenie pracy programu
 - b) **TAK** - kontynuowanie procesu, aż do niespełnienia warunku

7

Program pętla WHILE

```
program petla_while;
var i,j,k:integer;
begin
    i:=1;
    j:=1;
    k:=10;
    writeln('Program wypisuje liczby przy pomocy pętli');
    while (i <= 10) do
    begin
        writeln (i);
        i:=i+1;
    end;
    readln;
end.
```

8

Ćwiczenie

1. Zmień warunki początkowe i końcowe pętli

```
while (i <= k) do  
begin  
  writeln (i);  
  i:=i+1;  
end;
```
2. Sprawdź w jakich granicach można zmieniać wartości **k**.
 - a) Jakie maksymalne lub minimalne wartości może przybrać **k**?
 - b) Czy **k** może być ujemne? Jak wyświetlić liczby ujemne?
3. Korzystając z funkcji **Char(i)** napisać program wyświetlający tablicę kodową znaków ASCII. Ma zawierać znaki o kodach od 0 do 255. Wykorzystaj pętlę WHILE. Funkcja ma postać:
 - `Writeln (char (i));`
4. Zmień pętlę by wyświetlała liczby malejąco.

9

Przykład programu chodzącego w kółko

```
program w_kolko_chodzacy;  
var a:integer;  
begin  
  a:=1;  
  while (a>0) do  
  begin  
    {tu wpisuje sie kod programu}  
    writeln('Czy chcesz dalej kontynuowac?');  
    writeln('1-TAK, 0-NIE');  
    readln(a);  
  end;  
end.
```

10

Ćwiczenie

1. Zmień program pętla_WHILE – wpisz pętlę dekrementacyjną do programu.
2. Napisz program, który ma wczytywać liczbę **n** z zewnątrz i wypisywać liczby od **1** do **n** oraz od **n** do **1**. Wykorzystaj pętlę WHILE.
3. Napisz programu, który by umożliwił wczytanie kilku liczb i obliczenie ich sumy. Wykorzystaj pętlę WHILE.

11

Umieszczenie jednej pętli w drugiej

- Istnieje możliwość umieszczenia jednej pętli w drugiej.
- Wtedy operacje zawarte w środku wykonywane są tyle razy ile wynosi iloczyn poszczególnych pętli.

```
While (i<=10) do  
  While (j<=10) do  
    writeln (i + j);
```

12

Ćwiczenie

1. Napisz program, który wypisuje na ekranie zawartość tabliczki mnożenia w zakresie od 1 do 100. Wykorzystaj w tym celu pętlę WHILE.

13

14

Program suma kilku liczb

```
Program Suma_kilku_liczb;
var   I:Integer;
      Suma:Real;
      X:Real
Begin
  I:=1;           {Nadanie wartości początkowej I - licznik}
  Suma:=0;       {Nadanie wartości początkowej Suma}
  WHILE I<=6 do
  begin
    write ('podaj liczbę x=')
    Readln(x);
    Suma:=Suma+X;
    Writeln (' suma=',Suma:6:2);
    I:=I+1;      {W tym miejscu sterujemy pętlą przez}
  end;          {zwiększanie licznika I}
  Writeln (' końcowa suma=',Suma:6:2);
  Readln;
end.
```

15