

Pętle w Paskalu

Pętla REPEAT

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 REPEAT

- Pętla FOR służy do wykonania jakiegoś polecenia pewną ilość razy. Czasem jednak potrzeba nam, by pętla wykonywała **coś raz się ze względu na jakiś warunek**, a nie pewną ilość razy.
- Dlatego należy zastosować pętlę REPEAT. Wykonuje ona jakieś polecenia co najmniej jeden raz **dopóki warunek nie zostanie spełniony**.

- Ma ona postać:

```
REPEAT
    Operacja
UNTIL (Warunek);
```

- Przykład

```
Repeat
    AX:=BX
UNTIL (i >= 1000);
```

3

Pętla REPEAT

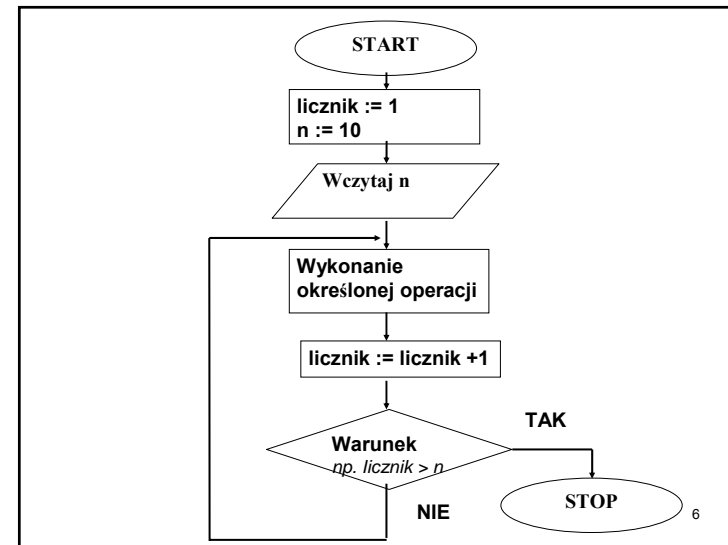
- REPEAT – UNTIL jest instrukcją pętli z kontrolowanym WEJŚCIEM. Jej działanie polega na wykonaniu instrukcji lub ciągu instrukcji i sprawdzeniu czy warunek nie został spełniony.
- Warunek jest sprawdzany **na końcu** wykonania serii instrukcji zawierających się pomiędzy **BEGIN** i **END**; aż do instrukcji UNTIL sprawdzającej warunek.
- Pętla wykona instrukcję co najmniej jeden raz – bo warunek jest sprawdzany na końcu.
- Warunek zwraca wartość logiczną, to znaczy
 - Jest spełniony - prawdę
 - Nie jest spełniony - fałsz.
- Seria instrukcji będzie wykonywana dopóki warunek jest spełniony.

4

Składnia pętli REPEAT

REPEAT	POWTARZAJ
BEGIN	WYKONUJ
instrukcja 1; instrukcja 2; instrukcja n; END;	instrukcja 1; instrukcja 2; instrukcja n; KONIEC
UNTIL (warunek)	DOPÓKI warunek nie jest spełniony

5



6

Ogólna zasada działania pętli

PĘTLA REPEAT

1. **Start** - uruchomienie programu
2. inicjalizacja wstępna zmiennych - *licznik:=1, n:=10*
3. **wczytaj n** - program wczytuje n
4. Wykonanie przewidzianej operacji.
5. Operacja zwiększania licznika o 1 (INKREMENTACJA)
6. *licznik > n* - **postawienie warunku**
 - a) **TAK** - spełnienie warunku (*licznik > n*) zatrzymanie programu
 - b) **NIE** - niespełnienie warunku (*licznik > n*) dalsza praca programu
7. Powrót do wykonywanych operacji (pkt 4)

7

Program pętli REPEAT

```
program petla_repeat;  
var i,j,k:integer;  
begin  
  i:=1;  
  j:=1;  
  k:=10;  
  writeln('Program wypisuje liczby przy pomocy pętli');  
  repeat  
    begin  
      writeln (i);  
      i:=i+1;  
    end  
  until (i > 10);  
  readln;  
end.
```

8

Ćwiczenie

1. Zmień warunki początkowe i końcowe pętli

```
repeat  
begin  
  writeln (i);  
  i:=i+1;  
end  
until (i > k);
```
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. 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;  
  repeat  
    begin  
      {tu wpisuje sie kod programu}  
      writeln('Czy chcesz dalej kontynuowac?');  
      writeln('1-TAK, 0-NIE');  
      readln(a);  
    end  
  until (a=0);  
end.
```

10

Ćwiczenie

1. Zmień program pętla_REPEAT – 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 REPEAT.
3. Napisz programu, który by umożliwił wczytanie kilku liczb i obliczenie ich sumy. Wykorzystaj pętlę REPEAT.

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.

```
Repeat  
  repeat  
    writeln (i + j)  
  until (j>10)  
until (i>10);
```

12

Ćwiczenie

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

13

14

15

16

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.
```