

Pętle w Paskalu

Pętla FOR

1

Pętla - definicja

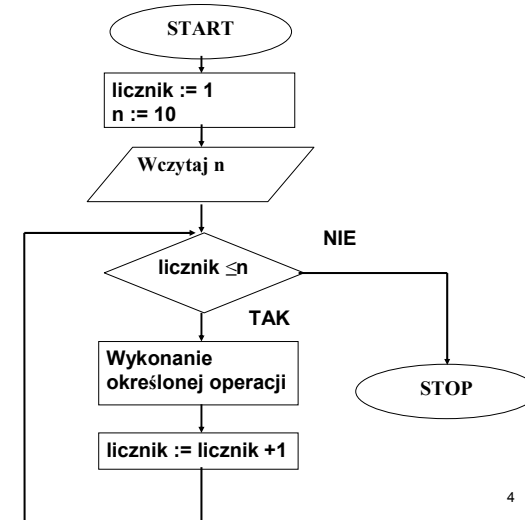
- **Pętla** to struktura języka programowania, która pozwala na cykliczne wykonanie wiele razy tej samej (lub podobnej) operacji.
- Pętla operuje na takich samych operacjach, danych, strukturach, obiektach czy procedurach.

2

Potrzeba stosowania pętli

- Mamy wykonać np. 1000 razy operację przypisania wartości do jakiejś zmiennej. Używając języków niestukturalnych np. assemblera, musielibyśmy taką instrukcję napisać 1000 razy (*ułatwimy sobie wprawdzie pracę używając polecenia **kopiuj – wklej**, ale to nie rozwiązuje problemu*).
- `Mov AX, BX` –
przenieś zawartość rejestru BX do rejestru AX.
- Korzystając z pętli upraszczamy sprawę:
`for i:=1 to 1000 do`
`AX:=BX;`
- Takie instrukcje nazywamy **INSTRUKCJAMI ITERACYJNYMI**.
 - Służą do ograniczenia cykli programowych, tj. wielokrotnego wykonywania pewnych sekwencji instrukcji.

3



4

Ogólna zasada działania pętli

PĘTLA FOR

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

5

Program pętla FOR

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

6

Ćwiczenie

1. Zmień warunki początkowe i końcowe pętli

```
for i:=j to k do
begin
  writeln (i);
end;
```
2. Sprawdź w jakich granicach można zmieniać wartości **j** i **k**.
 - a) Jakie maksymalne lub minimalne wartości mogą przybrać **j** i **k**?
 - b) Czy **j** i **k** mogą być ujemne?
 - c) Czy **j** może być większe od **k**? Co się wtedy wydarzy?
3. Korzystając z funkcji **Char(i)** napisz 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.

7

Pętla zliczająca w dół

- W języku Paskal istnieje możliwość zaimplementowania pętli liczącej w dół (*dekrementacyjnej*). Konstrukcja taka ma postać:

```
for i:=10 downto 1 do
  writeln (i);
```

8

Ćwiczenie

1. Zmień program pętla_FOR – 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**.

9

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.

```
for i:=1 do 10 do  
  for j:=1 do 10 do  
    writeln (i + j);
```

10

Ćwiczenie

1. Napisz program, który wypisuje na ekranie zawartość tabliczki mnożenia w zakresie od 1 do 100.

11

Kody programów

12

Program tablica ASCII

```
program tablica_ASCII;
var i,j:integer;
begin
  i:=1;
  writeln('Program wypisuje na ekranie zawartosc tablicy ASCII');
  for i:=0 to 255 do
    write (char(i),' ');
  readln;
end.
```

13

Program tabliczka mnozenia

```
program tabliczka_mnozenia;
var i,j:integer;
begin
  i:=1;
  j:=1;
  writeln('Program wypisuje na ekranie tabliczke mnozenia');
  for i:=1 to 10 do
    begin
      for j:=1 to 10 do
        begin
          write (i*j:3,' ');
        end;
        writeln;
      end;
    end;
  readln;
end.
```

14