

# Java

---

wprowadzenie  
instrukcja iteracyjna - FOR

# Czego się dowiesz?

- Czym jest pętla FOR?
- Jak ją stworzyć?
- Jakie jest jej zastosowanie?
- Kiedy zamiast WHILE użyjemy FOR?

# FOR

For jest kolejną instrukcją iteracyjną, tzw. pętlą. Podobnie jak while i do-while umożliwia powtarzanie wykonywania określonego kodu. Różni się od pozostałych konstrukcją i zastosowaniem.

Pętla FOR może zastąpić dowolny zapis pętli WHILE i na odwrót. Pętla for znajduje zastosowanie w sytuacjach, gdy operujemy na indeksach. Jest ona dużo czytelniejsza od podobnego rozwiązania w pętli while.

# Definicja

Rozpocznijmy od definicji bardzo ogólnej, która pozwala spojrzeć możliwie najszerzej na pętlę FOR.

```
for( blok1 ; blok2; blok3 ){  
    instrukcje;  
}
```

blok1 - wykonywany tylko jeden raz, przed uruchomieniem pętli

blok2 - warunek zakończenia pętli, musi zwracać true albo false

blok3 - wykonywany na końcu każdego obiegu pętli

instrukcje - wykonywane, gdy blok2 jest prawdziwy

# Kolejność wykonywania

```
for( blok1 ; blok2; blok3 ){  
    instrukcje;  
}
```

- 1.wykonanie bloku1
- 2.sprawdzenie warunku (blok2)
  - a.jeśli niepoprawny, to koniec
  - b.jeśli poprawny, to:
    - i.wykonanie instrukcji
    - ii.wykonanie blok3
    - iii.powrót do punktu 2

# Kolejność wykonywania

```
for( blok1 ; blok2; blok3 ){  
    instrukcje;  
}
```

1. **blok1**

2. **blok2**

a.jeśli nie to koniec

b.jeśli tak to

i. **instrukcje**

ii. **blok3**

iii. powrót do pkt. 2

# Opcjonalność bloków

```
for( blok1 ; blok2; blok3 ){  
    instrukcje;  
}
```

Każdy z bloków jest opcjonalny, nie musisz go podawać.

# Przykład 1

Co stanie się w sytuacji, gdy zapiszemy taką pętlę?

```
for(;;){}
```

Pętla będzie w nieskończoność wykonywała "nic". Nigdy się nie zakończy, domyślnie warunek będzie cały czas poprawny.



# Przykład 2

Co stanie się w sytuacji, gdy zapiszemy taką pętlę?

```
for(;;){  
    System.out.println("Hej!");  
}
```

Pętla będzie w nieskończoność wyświetlała słowo "Hej!". Nigdy się nie zakończy, domyślnie warunek będzie cały czas poprawny.

# Przykład 3

Zamiana pętli WHILE na pętlę FOR.

## WHILE

```
int i = 0;
while(i < 10) {
    System.out.println(i);
    i++;
}
```

## FOR

```
for(int i = 0; i < 10; i++){
    System.out.println(i);
}
```

# Przykład 3

## WHILE

```
int i = 0;
while(i < 10) {
    System.out.println(i);
    i++;
}
```

## FOR

```
for(int i = 0; i < 10; i++){
    System.out.println(i);
}
```

Do bloku1 trafiła deklaracja oraz inicjalizacja zmiennej, która zawsze **wykonuje się jeden raz**. Nie możemy dwukrotnie deklarować tej samej zmiennej.

Do bloku2 trafił identyczny **warunek** jak w pętli while.

Do bloku3 trafiła iteracja, która zawsze **wykonuje się na końcu pętli**.

# Definicja

Ze względu na powyższe, FOR bardzo często będzie występowało w takiej formie:

```
for( blok_inicjalizacyjny ; warunek ; blok_inkrementacji ){  
    instrukcje;  
}
```

blok1 -> blok\_inicjalizacyjny

blok2 -> warunek

blok3 -> blok\_inkrementacji

# Przykład 4 (nietypowy)

Nie zapominaj, że FOR jest czymś więcej.

Co się stanie, gdy... ?

```
for(System.out.println("Hej!"); true; System.out.println("*")){}
```

Działanie pętli się nie zmieni. Blok1, wykona się jeden raz, a więc jeden raz zostanie wyświetlone "Hej!". Blok2 (warunek) jest zawsze prawdziwy, więc pętla nigdy się nie zakończy. Nie wykona się żadna instrukcja, ale... na końcu każdego obiegu zostanie wyświetlony Blok3 czyli gwiazdka.

Czyli zobaczysz "Hej!" a potem same gwiazdki, każda w nowej linii.

# Podsumowanie

W wielu przypadkach pętla FOR zastąpi pętlę WHILE i będzie czytelniejsza.

Pętla WHILE nadal będzie potrzebna, gdy nie będziemy "iterować". Na tę chwilę przede wszystkim działamy na indeksach, więc w większości przypadków wybór padnie na FOR.

# Zadanie 1

Wyświetl liczby od 91 do -7.

# Zadanie 2

Wyświetl liczby od 62349 do 62392 podzielne przez 2.



# Zadanie 3

Wyświetl litery alfabetu od 'a' do 'z'.

# Zadanie 4

Wyświetl dowolną tablicę dwuwymiarową.

# Zadanie 5

Posortuj dowolną tablicę liczb całkowitych, od najmniejszej do największej.

# Zadanie 6\*

Dla dowolnej listy ocen, oblicz ich średnią.

np.

dla ocen  $\{1, 3, 5, 4, 5\}$

średnia 3.6

dla ocen  $\{3, 4, 4, 4, 4, 3, 3, 5\}$

średnia 3.75

\* wykonaj samodzielnie

# Zadanie 7\*

Dla dowolnej tablicy znaków policz, ile występuje tam liter 'a' lub 'A'.

\* wykonaj samodzielnie

# Zadanie 8\*

Dla dowolnych liczb podaj ich:

- sumę
- iloczyn
- znajdź najmniejszą
- znajdź największą
- posortuj malejąco
- oblicz medianę

\* wykonaj samodzielnie

# Zadanie 9\*

Dla podanego słowa obróć go. Korzystaj z typu char.

np.

Jurek -> keruj

mama -> amam

tunel -> lenut

\* wykonaj samodzielnie