

# Ćwiczenie 2

## Iteracja i rekurencja

### 1 Wprowadzenie

#### 1.1 Operatory relacji

Operatory relacji służą do porównywania argumentów. Wynikiem ich działania jest wartość logiczna *true* albo *false*.

Operator	Działanie
==	równy
!=	różny
<	mniejszy
>	większy
<=	mniejszy lub równy
>=	większy lub równy

#### 1.2 Operatory logiczne

Argumentami operatorów logicznych mogą być zmienne lub wyrażenia typu logicznego.

##### 1.2.1 Operatory logiczne jednoargumentowe

Operator	Działanie
!	negacja

$x$	$!x$
false	true
true	false

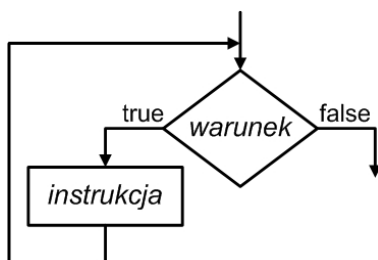
##### 1.2.2 Operatory logiczne dwuargumentowe

Operator	Działanie
&&	koniunkcja (iloczyn logiczny)
	alternatywa (suma logiczna)

$x$	$y$	$x \&\& y$	$x \ \  y$
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

## 1.3 Instrukcje iteracyjne

### 1.3.1 Instrukcja WHILE

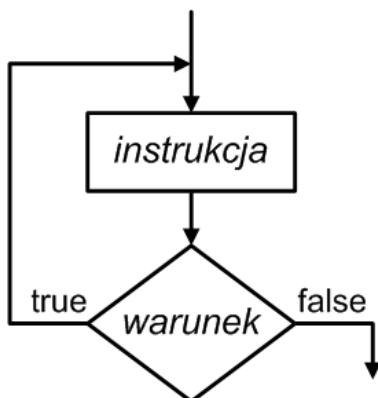


Składnia:

```
while(warunek)
instrukcja
```

- Wartością warunku powinna być wartość logiczna *true* albo *false*.
- Dopóki warunek jest prawdziwy (*true*), instrukcja jest wykonywana. Gdy warunek staje się fałszywy (*false*) sterowanie przenoszone jest do kolejnych instrukcji po instrukcji WHILE.
- Instrukcja może być instrukcją złożoną.

### 1.3.2 Instrukcja DO-WHILE

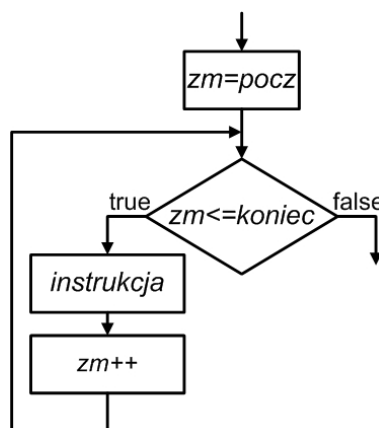


Składnia:

```
do
{
instrukcja
}
while(warunek);
```

- Wartością warunku powinna być wartość logiczna *true* albo *false*.
- Dopóki warunek jest prawdziwy (*true*), następuje powrót na początek pętli i wykonanie instrukcji. Gdy warunek staje się fałszywy (*false*) sterowanie przenoszone jest do kolejnych instrukcji po instrukcji DO-WHILE.
- Instrukcja jest wykonana co najmniej jeden raz.
- Instrukcja może być instrukcją złożoną.

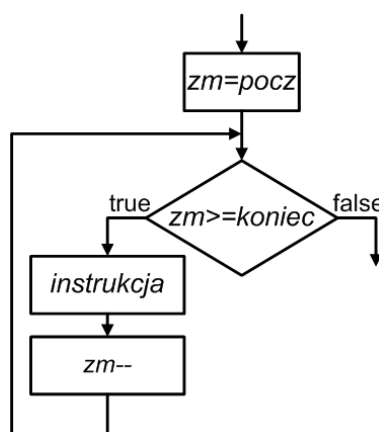
## 1.3.3 Instrukcja FOR



Składnia:

```
for(zm=pocz; zm<=koniec; zm++)
instrukcja
```

- Zmienna sterująca pętli *zm* jest typu całkowitego.
- Zmiennej *zm* przypisywana jest wartość *pocz*. Dopóki wartość zmiennej jest mniejsza lub równa wartości *koniec*, instrukcja jest wykonywana. Za każdym razem wartość zmiennej *zm* zwiększana jest o 1. Gdy wartość zmiennej *zm* staje się większa od wartości *koniec* sterowanie jest przenoszone do kolejnych instrukcji po instrukcji FOR.
- Instrukcja może być instrukcją złożoną.



Składnia:

```
for(zm=pocz; zm>=koniec; zm--)
instrukcja
```

- Zmienna sterująca pętli *zm* jest typu całkowitego.
- Zmiennej *zm* przypisywana jest wartość *pocz*. Dopóki wartość zmiennej jest większa lub równa wartości *koniec*, instrukcja jest wykonywana. Za każdym razem wartość zmiennej *zm* zmniejszana jest o 1. Gdy wartość zmiennej *zm* staje się mniejsza od wartości *koniec* sterowanie jest przenoszone do kolejnych instrukcji po instrukcji FOR.
- Instrukcja może być instrukcją złożoną.

## 2 Zadania

### 2.1

Napisz metodę dodającą liczby wprowadzane przez użytkownika aż do momentu wprowadzenia liczby 0.

### 2.2

Napisz metody iteracyjną i rekurencyjną obliczania wartości wyrażenia  $2^n$  dla podanej liczby naturalnej  $n$ .

### 2.3

Napisz metody iteracyjną i rekurencyjną obliczania wartości silni dla podanej liczby naturalnej  $n$ .

### 2.4

Napisz metody wyznaczające wartości szeregów:

- $y = \sum_{i=1}^{30} \frac{i}{i+2}$
- $y = \sum_{j=1}^5 2j$
- $y = \sum_{i=1}^{20} \sum_{j=1}^{10} \frac{i+j}{2^i}$
- $y = \sum_{j=1}^{40} \sum_{k=1}^{20} j + 2k$

### 2.5

Napisz metody rekurencyjne wyznaczające  $n$ -te wyrazy ciągów:

- $a(x) = \begin{cases} 2 & \text{dla } x = 0 \\ a(x-1) + 3x & \text{dla } x > 0 \end{cases}$
- $b(x) = \begin{cases} 1 & \text{dla } x = 0 \\ 3 & \text{dla } x = 1 \\ 2b(x-1) + 3b(x-2) & \text{dla } x > 1 \end{cases}$

### 2.6

Napisz metodę, która z podawanych przez użytkownika dziesięciu liczb całkowitych wyznaczy liczbę największą.

### 2.7

Napisz metodę, która dla podawanych przez użytkownika pięciu liczb rzeczywistych wyznaczy ilość liczb z przedziału  $[-1, 1]$ .

## 2.8

Napisz metodę, która wśród liter wprowadzanych przez użytkownika zlicza ilość dużych liter. Zakończenie wprowadzania liter powinno być zasygnalizowane przez użytkownika wprowadzeniem znaku innego niż litera.