

## Lista 5

Termin oddania: wtorek – 02.04.2019

**Zadanie:** Napisz program, który:

1. czyta wyrażenie,
2. zamienia je na postać ONP (odwrotna notacja polska),
3. wylicza wyrażenie i wyświetla jego wynik,
4. Wyświetla stosowny komunikat, jeśli w trakcie wyliczeń natknęło się na błąd
5. Analizowane są jedynie operatory: +, -, \*, /, %
6. Operandami mogą być tylko liczby całkowite

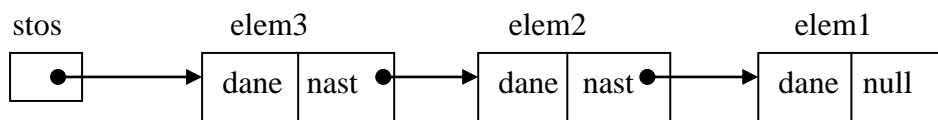
**Przykład:**  $(2+3)*2 = 2\ 3 + 2\ * = 10$

**W programie należy zdefiniować**

### ♦ Interfejs generyczny Stack

- empty() – zwraca *true*, jeśli stos jest pusty,
- push() – umieszcza znak na szczycie stosu i zwraca referencje do niego,
- pop() – zwraca znak ze szczytu stosu i usuwa go ze stosu,
- peek() – zwraca znak ze szczytu stosu bez jego zdejmowania.

♦ Klasa Stos implementująca interfejs, stos – implementacja jako lista wiązana jednokierunkowa.



## Konwersja z zapisu infiksowego na ONP

a+b                    a b +  
 operatory:        +, -, \*, /, %  
 priorytety:        1, 1, 2, 2, 2

Klasa InfixtoPostfix

Atrybuty:        stos operatorów,

postfix - wyrażenie ONP

Metody:        public String convert (String infix)  
                  private void processOperator(char op)  
                  private int precedence(char op)                    //zwraca priorytet operatora  
                  private boolean isOperator(char char)            //zwraca true, jeśli char operatorem

Metoda convert ()

1. inicjalizacja pustego stosu
2. inicjalizacja pola na wyrażenie ONP
3. while (jest znak na wejściu)
4. czytaj znak
5. if znak to operand

6. dodaj go do wyrażenia ONP (`postfix`)
7. `else if` znak to operator
8. wywołaj metodę `processOperator()`
9. `else`
10. wyświetl błąd
11. zdejmij operatory ze stosu i dodaj je do ONP

Metoda `processOperator()`

1. `if` stos operatorów pusty
2. umieść operator na stosie
- `else`
3. pobierz operator ze szczytu stosu (bez jego usuwania) do zmiennej `topOp`
4. `if` priorytet aktualnego operatora jest większy od priorytetu `topOp`
5. umieść aktualny operator na stosie
- `else`
6. `while` stos nie jest pusty i priorytet aktualnego operatora jest mniejszy lub równy priorytetowi `topOp`
7. zdejmij `topOp` ze stosu i dodaj go do zmiennej `postfix`
8. `if` stos operatorów nie jest pusty
9. pobierz operator ze szczytu stosu (bez jego usuwania) do zmiennej `topOp`
10. umieść aktualny operator na stosie.

## Konwersja wyrażen z nawiasami

(a+b) a b +

operatory: +, -, \*, /, %, (, )

priorytety: 1, 1, 2, 2, 2, -1, -1

Modyfikacja:

Metoda `processOperator()`

1. `if` stos operatorów pusty lub `op == '('`
2. umieść operator na stosie
- `else`
- ...
6. `while` stos nie jest pusty i priorytet aktualnego operatora jest mniejszy lub równy priorytetowi `topOp`
7. zdejmij `topOp` ze stosu
- `if (topOp=='(' { break; //- wyjście z pętli`
- `}`
- i dodaj go do zmiennej `postfix`
- ...
10. `if (op != ')')` umieść aktualny operator na stosie.