

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №7
з дисципліни
«Об’єктно орієнтоване програмування»

Виконав:
студент групи КН-108
Кіт В.Д
Викладач:
Гасько Р.Т.

Львів – 2017 р.

Створення застосування для обрахунку виразів що задано в формі польського інверсного запису

Код:

```
package com.tasks7.rpn;

import java.util.Deque;
import java.util.LinkedList;

public class Application {

    private final static String DELIM = " ";

    public static double parse(String rpnString) {
        if (rpnString == null || rpnString.isEmpty()) {
            throw new RPNParserException();
        }
        Deque<Double> stek = new LinkedList<Double>();
        String[] result = rpnString.split(DELIM);
        for (int i = 0; i < result.length; i++) {
            if (isNumber(result[i])) {
                stek.push(new Double(result[i]));
            } else if (isOperator(result[i])) {
                if (stek.size() < 2) {
                    throw new RPNParserException();
                }
                switch (result[i]) {
                    case "+":
                        stek.push(new Double(stek.pop() + stek.pop()));
                        break;
                    case "-":
                        stek.push(new Double(-stek.pop() + stek.pop()));
                        break;
                    case "/":
                        if (stek.peek() == 0){
                            throw new ArithmeticException();
                        }
                        stek.push(new Double(1/stek.pop() * stek.pop()));
                        break;
                    case "*":
                        stek.push(new Double(stek.pop() * stek.pop()));
                        break;
                }
            } else {
                throw new RPNParserException();
            }
        }
        if (stek.size() != 1) {
            throw new RPNParserException();
        }
        return stek.pop();
    }

    private static boolean isNumber(String string) {
```

```

    if (string == null)
        return false;
    return string.matches("^-?\\d+(\\.\\d+)?$");
}

private static boolean isOperator(String string) {
    if (string == null)
        return false;
    return string.matches("[+/*]{1}");
}

public static void main(String[] args) {

    if (args == null || args.length == 0) {
        throw new RPNParserException();
    }
    System.out.println(parse(args[0]));
}
}

```

