

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Лабораторна робота №7

з курсу:

“ОБ’ЄКТНО ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ”

Виконав:
ст. гр. КН-110
Шевчук Ігор
Прийняв:
Гасько Р.Т.

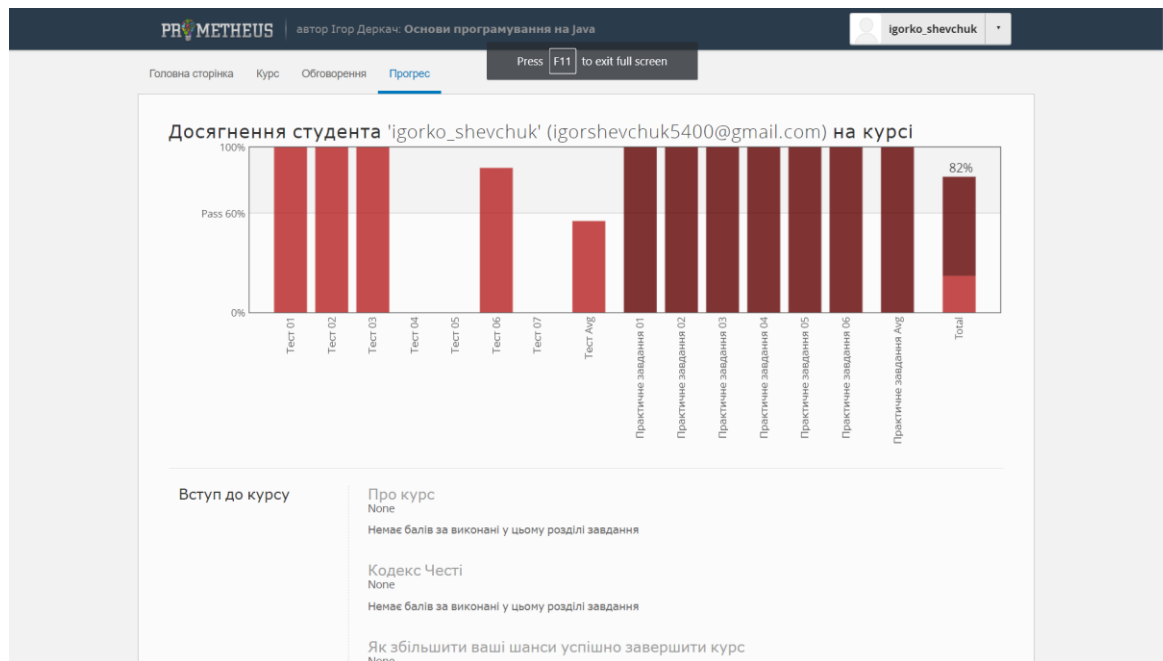
Львів – 2018 р.

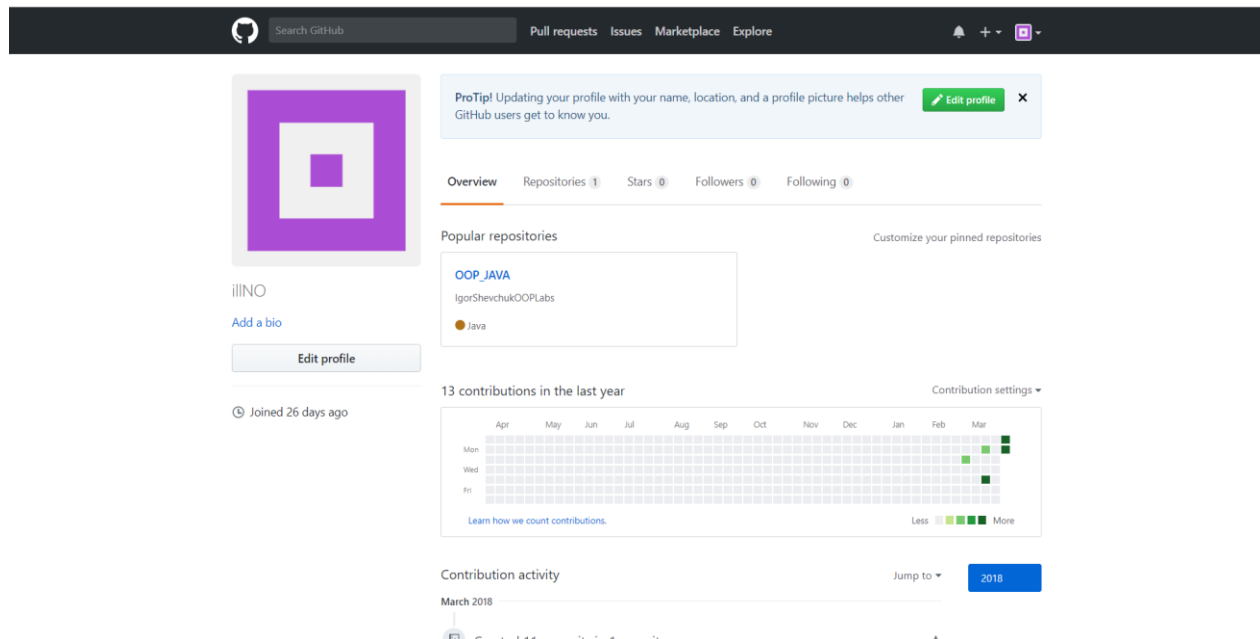
Лабораторна робота №7

Завдання: Опрацювати 7 тиждень на prometheus.org

Виконання роботи

GitHub link - <https://github.com/illNO>





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

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Main {

    public static double parse(String rpnString) {

        // new stack
        Deque<Double> stack = new LinkedList<>();

        // match for correct string
        Pattern p = Pattern.compile("^([0-9+\\-*/\\.\\s]+)$");
        Matcher m = p.matcher(rpnString);
        if (!m.matches()) {
            throw new RPNParserException();
        }

        // split sting to string array
        String[] inSplit = rpnString.split(" ");

        //
        try {
            Double.parseDouble(inSplit[0]);
            Double.parseDouble(inSplit[1]);
        } catch (Exception e) {
            throw new RPNParserException();
        }
    }
}
```

```

    }

    // last string must be sign
    p = Pattern.compile("[+\\-*/]$");
    m = p.matcher(inSplit[inSplit.length-1]);
    if (!m.matches()) {
        throw new RPNParserException();
    }

    // rpn calculating
    for (String step: inSplit) {
        switch (step) {
            case "+":
                stack.push(stack.pop() + stack.pop());
                break;
            case "-":
                double sub1 = stack.pop();
                double sub2 = stack.pop();
                stack.push(sub2 - sub1);
                break;
            case "*":
                stack.push(stack.pop() * stack.pop());
                break;
            case "/":
                double div1 = stack.pop();
                double div2 = stack.pop();

                if (Double.isInfinite(div1) || Double.isNaN(div1) || div1 == 0) {
                    throw new ArithmeticException();
                } else {
                    double result = div2 / div1;

                    stack.push(result);
                }
                break;
            default:
                stack.push(Double.parseDouble(step));
        }
    }

    // return result
    return stack.pop();
}

public static void main(String[] args) {
    System.out.println(parse("10 20 + 30 40 + *"));
}

```

}

Висновок: на цій лабораторній роботі я створив застосування для обрахунку виразів що задано в формі польського інверсного запису