UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería

Profesor: M.I. Marco Antonio Martínez Quintana

Estructura de Datos y Algoritmos I

Actividad asíncrona #5 | Calculadora

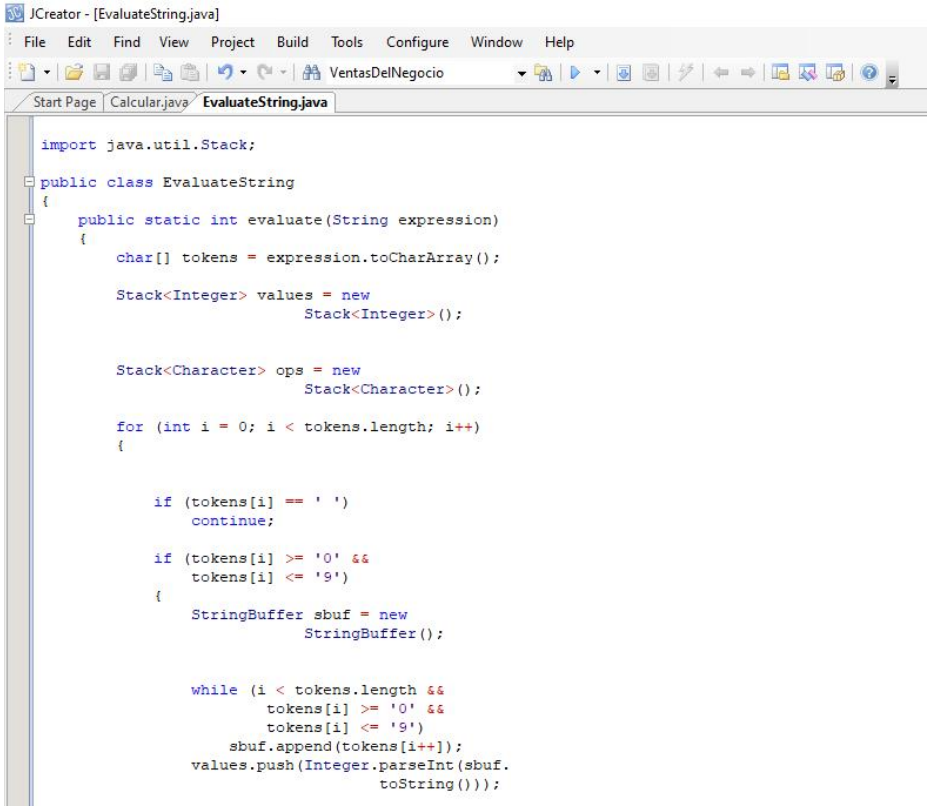
Alumna: Pineda Cruz Tania

No. de lista

Grupo: 15

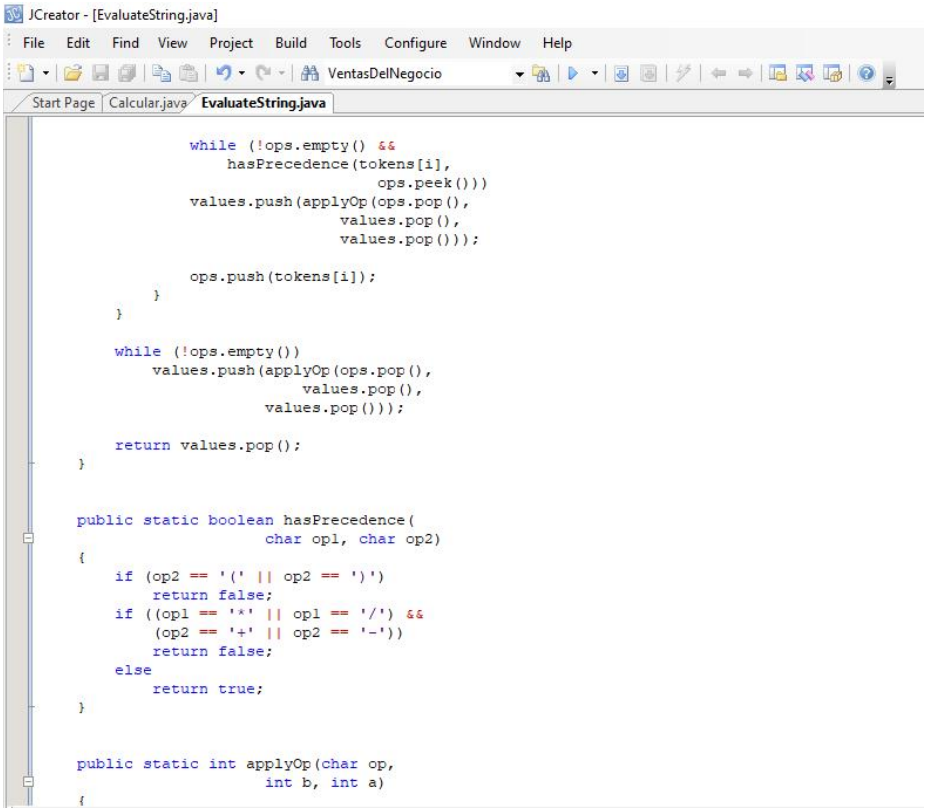
18/06/2021

El misterio de las calculadoras (algoritmo)

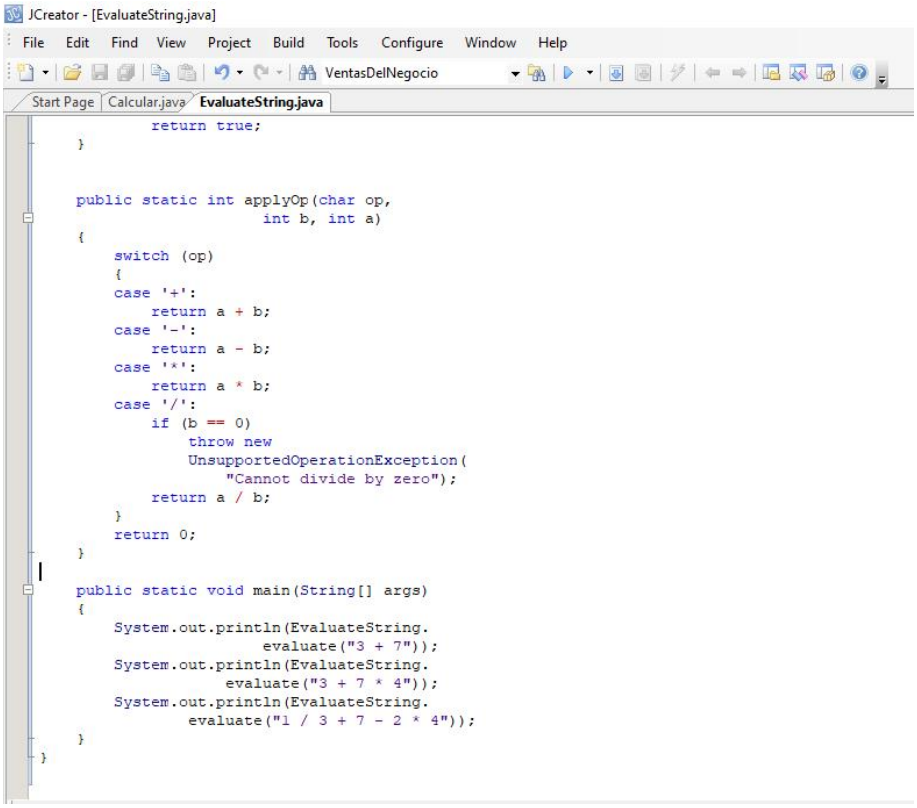



```
import java.util.Stack;

public class EvaluateString
{
    public static int evaluate(String expression)
    {
        char[] tokens = expression.toCharArray();

        Stack<Integer> values = new
                          Stack<Integer>();


        Stack<Character> ops = new
                          Stack<Character>();

        for (int i = 0; i < tokens.length; i++)
        {


            if (tokens[i] == ' ')
                continue;

            if (tokens[i] >= '0' &&
                tokens[i] <= '9')
            {
                StringBuffer sbuf = new
                          StringBuffer();


                while (i < tokens.length &&
                        tokens[i] >= '0' &&
                        tokens[i] <= '9')
                    sbuf.append(tokens[i++]);
                values.push(Integer.parseInt(sbuf.
                                  toString()));
```


```
                while (i < tokens.length &&
                        tokens[i] >= '0' &&
                        tokens[i] <= '9')
                    sbuf.append(tokens[i++]);
                values.push(Integer.parseInt(sbuf.
                                  toString()));

                i--;
            }

            else if (tokens[i] == '(')
                ops.push(tokens[i]);


            else if (tokens[i] == ')')
            {
                while (ops.peek() != '(')
                values.push(applyOp(ops.pop(),
                                  values.pop(),
                                  values.pop()));
                ops.pop();
            }

            else if (tokens[i] == '+' ||
                    tokens[i] == '-' ||
                    tokens[i] == '*' ||
                        tokens[i] == '/')
            {

                while (!ops.empty() &&
                    hasPrecedence(tokens[i],
                                  ops.peek()))
                values.push(applyOp(ops.pop(),
                                  values.pop(),
                                  values.pop()));

                ops.push(tokens[i]);
```




```
                while (!ops.empty() &&
                    hasPrecedence(tokens[i],
                                  ops.peek()))
                values.push(applyOp(ops.pop(),
                                    values.pop(),
                                    values.pop()));

                ops.push(tokens[i]);
            }
        }

        while (!ops.empty())
            values.push(applyOp(ops.pop(),
                                values.pop(),
                        values.pop()));

        return values.pop();
    }


    public static boolean hasPrecedence(
                        char op1, char op2)
    {
        if (op2 == '(' || op2 == ')')
            return false;
        if ((op1 == '*' || op1 == '/') &&
            (op2 == '+' || op2 == '-'))
            return false;
        else
            return true;
    }


    public static int applyOp(char op,
                        int b, int a)
    {
```




```
            return true;
    }


    public static int applyOp(char op,
                        int b, int a)
    {
        switch (op)
        {
        case '+':
            return a + b;
        case '-':
            return a - b;
        case '*':
            return a * b;
        case '/':
            if (b == 0)
                throw new
                UnsupportedOperationException(
                    "Cannot divide by zero");
            return a / b;
        }
        return 0;
    }

    public static void main(String[] args)
    {
        System.out.println(EvaluateString.
                        evaluate("3 + 7"));
        System.out.println(EvaluateString.
                    evaluate("3 + 7 * 4"));
        System.out.println(EvaluateString.
                evaluate("1 / 3 + 7 - 2 * 4"));
    }
}
```

Operaciones por resolver

- "3+7"
- "3+7*4"
- "1/3+7-2*4"