

ASSIGNMENT 5.1 -62070501064 ONWIPA KUJAROENPAISAN-

SOURCE CODE

```

1 package ass5;
2
3 import java.util.*;
4
5 public class ass5 {
6
7     public static String Transform(String x) { //function to split the operator
8         x = x.replace("+", " + ");
9         x = x.replace("-", " - ");
10        x = x.replace("*", " * ");
11        x = x.replace("/", " / ");
12        x = x.replace("^", " ^ ");
13        x = x.replace("(", " ( ");
14        x = x.replace(")", " ) ");
15        x = x.replaceAll("\\s+", " "); //change the space
16        x = x.trim(); //cut the space before and behind
17        return x;
18    }
19
20    public static boolean isNum(String str) { //function to check the number
21        boolean check = true;
22        try {
23            Double.parseDouble(str);
24        } catch (Exception e) {
25            check = false;
26        }
27        return check;
28    }
29
30    public static int check_Function(String str) { //function to check string in form math function
31        String[] myfunc = {"", "sin", "cos", "tan", "cosec", "sec", "cot", "asin", "acos", "atan", "sqrt", "log", "exp", "abs"};
32        int ans = 0;
33        for(int i = 1; i < myfunc.length; i++) {
34            if(myfunc[i].equalsIgnoreCase(str))
35                ans = i;
36        }
37        return ans;
38    }
39
40    public static boolean isConstant(String str) { //function to check math constant
41        String[] myconst = {"pi", "e", "ans"};
42        boolean ans = false;
43        for(int i = 0; i < myconst.length; i++) { //and count the number
44            if(myconst[i].equalsIgnoreCase(str))
45                ans = true;
46        }
47        return ans;
48    }
49
50    public static int check_group(String str) { //separate operator in group and return by number
51        if(isNum(str))
52            return 1;
53        else if(isConstant(str))
54            return 1;
55        else if(str.matches("[+ ]"))
56            return 2;
57        else if(str.matches("[ / * ]"))
58            return 3;
59        else if(str.equals("^"))
60            return 4;
61        else if(str.equals("("))
62            return 5;
63        else if(str.equals("("))
64            return 7;
65        else if(str.equals("("))
66            return 8;
67        else if(check_Function(str) > 0)
68            return 6;

```

```

69     else
70         return 0; //if its in unknown group then return 0
71     }
72
73     public static int check_state(String[] token, int count) {
74         int state = 0, next = 0, b = 0, i;
75         for(i = 0; i < count && state >= 0; i++) {
76             state = next; //declare state to be new value
77             next = check_group(token[i]);
78             if(next >= 2 && next <= 4) next = 2; //make state of +*/^ be state number 2
79             if(next == 0) state = -1; //unknown state is error
80             if(next == 7) b++; //check (
81             else if(next == 8) b--; //check )
82             if(b < 0)
83                 state = -1; //if its ) more than or before ( then show error
84             else if(state == 0 && (next == 8 /*|| next == 2*/))
85                 state = -1; //error if its begin with )
86             else if(state == 1 && (next == 1 || next == 6 || next == 7))
87                 state = -1; //error if its variant fn or ( after number
88             else if(state == 2 && (next == 2 || next == 8))
89                 state = -1; //error if its operator or ) after operator
90             else if(state == 5 && (next == 2 || next == 8))
91                 state = -1; //error if its operator or ) after sign
92             else if(state == 6 && (next != 7))
93                 state = -1; //error if its fn but not after by (
94             else if(state == 7 && (next == 8 /*|| next == 2*/))
95                 state = -1; //error if its be like ()
96         } //loop til end
97         if(b != 0 || state == -1) //if theres not couple of () or state is error
98             return -1;
99         else
100             return next; //return last state
101     }
102
103
104     public static void main(String[] args) {
105         String str;
106         Scanner in = new Scanner(System.in);
107         String[] token;
108         int i, code;
109         int count = 0;
110         double num;
111         do {
112             System.out.printf("expression> ");
113             str = in.nextLine(); //read string 1 line
114             str = str.toLowerCase(); //change input to lower character
115             str = Transform(str); //send input to split the operator
116             token = str.trim().split("\\s+"); //keep input that split by space in token
117             count = token.length; //count the token
118             i = check_state(token, count); //send to check state
119             if((i == -1 || i == 2 || i == 4 || i == 5 || i == 7) && !token[0].equalsIgnoreCase("end") && !token[0].equalsIgnoreCase("help")) {
120                 System.out.println("answer> error");
121             }
122             else if(!token[0].equalsIgnoreCase("end") && !token[0].equalsIgnoreCase("help")) {
123                 System.out.println("answer> OK");
124             }
125             else if(token[0].equalsIgnoreCase("help")) {
126                 System.out.println("answer> token = sin, cos, tan, asin, acos, atan, sqrt, log, exp, abs, +, -, *, /, ^, (, ), pi, ans");
127             }
128         } while(!token[0].equalsIgnoreCase("end")); //still wrong
129         System.out.printf("End Program");
130         System.out.printf("Program written by ONWIPA KUJAROENPAISAN 62070501064");
131     }
132
133 }
134

```

เมื่อรันโปรแกรม จะเริ่มโดยการให้ผู้ใช้งานป้อนข้อมูลที่ต้องการ เป็นลักษณะการคำนวณเชิงคณิตศาสตร์ เมื่อผู้ใช้งานเรียบร้อยแล้ว โปรแกรมจะอ่านสิ่งที่ป้อนเข้ามาและเก็บในตัวแปร str จากนั้นก็จะเปลี่ยนให้เป็นตัวพิมพ์เล็ก และส่งไปฟังก์ชัน Transform เพื่อเว้นวรรคให้กับสัญลักษณ์ทางคณิตศาสตร์ จากนั้นจะเก็บแต่ละตัวที่เว้นวรรคไว้เป็นตัวแปร token และนับจำนวน token เอาไว้ และจะส่งไป check state โดยจะนำ token ไปแบ่งกลุ่มต่างๆก่อน เช่น ตัวเลข ค่าคงที่ หรือสัญลักษณ์ต่างๆ จากนั้นก็ส่งกลับมาเช็คในฟังก์ชัน

check_state ว่าการเรียงประโยคที่ผู้รับป้อนมานั้น error หรือไม่ และ return ผลลัพธ์กลับไป main และนำไปแสดงผลของการเช็ค state ออกมาเป็นประโยคต่างๆ

TEST CASE

```
expression> 2.5+pi
answer> OK
expression> 2.5 pi +
answer> error
expression> -2.54 + 1.68 ( 2 + 3 )
answer> error
expression> -2.5 - sin ( -30 )
answer> OK
expression> sin(cos(tan(asin(acos(atan(log(sqrt(exp(abs(ans)))))))))))
answer> error
expression> sin(cos(tan(asin(acos(atan(log(sqrt(exp(abs(ans)))))))))))
answer> OK
expression> 2+3-((4))
answer> OK
expression> 2+3-(4/5*)
answer> error
expression> 2+3-(4/5*6+())
answer> error
expression> 2+3)+4/5*6+((7)
answer> error
expression> -1+2-3*4/5^6
answer> OK
expression> help
answer> token = sin, cos, tan, asin, acos, atan, sqrt, log, exp, abs, +, -, *, /, ^, (, ), pi, ans
expression> end
End Program
```

ปัญหาที่พบในการทำ ASSIGNMENT

- ยังไม่คล่องภาษาจาวา
- บางอย่างต้องเปิดหาในอินเทอร์เน็ต บางอย่างก็ถามจากเพื่อน

Self-Assessment : 3 เข้าใจแต่มีปัญหบางช่วงยังต้องขอความช่วยเหลือ