

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package deber2;

import java.util.Iterator;
import java.util.LinkedList;
import java.util.Stack;

/**
 *
 * @author RICHARD
 */
public class Expresion {

    public Expresion() {
    }
    static boolean verificaParentesis(String expresion) {
        Stack p1=new Stack();
        int tam=expresion.length();
        for(int i=0; i<tam;i++){
            char a =expresion.charAt(i);
            if(a=='('){
                p1.push("(");
            }
            if(a==')'){
                p1.pop();
            }
        }
        return p1.isEmpty();
    }

    static String InFixToPostFix(String expresion) {

        Stack<Character> p1,p2,p3;
        char a,b,c;
        p1=new Stack();
        p2=new Stack();
        p3=new Stack();
        String total="",total1="";

        int tam=expresion.length();

        for(int i=0; i<tam;i++){

            a =expresion.charAt(i);
            if(a=='('){
                p2.push('(');
            }
            else if(a==')'){
                p2.pop();
                if(!p1.isEmpty()){
                    total=total+p1.peek();
                    p1.pop();
                }
                if(!p3.isEmpty()){
                    total=total+p3.peek();
                    p3.pop();
                }
            }
        }
    }
}

```

```

        }
    }else if((expresion.charAt(i-1)=='')&&(expresion.charAt(i+1)=='')){
        p3.push(a);
    }else if(Character.isLetter(a)){
        total=total+a;
    }else{
        if(p1.isEmpty()){
            p1.push(a);

        }else {
            if((a=='+'||a=='-')){
                total=total+p1.peek();
                p1.pop();
            }
            p1.push(a);

        }else {
            if((p1.peek()=='+')||(p1.peek()=='-')){
                p1.push(a);
            }else{
                total=total+p1.peek();
                p1.pop();
                p1.push(a);
            }
        }
    }
    if(i+1==tam){
        total=total+p1.peek();
        p1.pop();
    }
}
return total;
}

static String evaluar(LinkedList<Variable> variables, String expresion){
    Variable vs=new Variable();
    float x=0,y=0;
    Stack<Character> lista1 =new Stack();
    float result=0;
    int tam =expresion.length();
    for(int i=0;i<tam;i++){
        char a=expresion.charAt(i);
        if(lista1.size()==2){
            char b=lista1.peek();
            System.out.println(lista1);
            System.out.println(b);
            lista1.pop();
            char c=lista1.peek();
            System.out.println(lista1);
            lista1.pop();
            System.out.println(lista1);
            Iterator<Variable> var=variables.iterator();
            if(c=='@'){
                y=result;
            }
            while(var.hasNext()){
                Variable s=var.next();

```

```

        if(b==s.nombre){
            x=x+s.valor;
        }
        if(c==s.nombre){
            y=y+s.valor;
        }
    }

    if(Character.isLetter(a)){
        lista1.add(a);
    }else if(a=='+' ){
        result=x+y;
        lista1.push('0');
        x=0;
    } else if(a=='-' ) {
        result=y-x;
        lista1.push('0');
        x=0;
    }else if(a=='*' ){
        result=y*x;
        lista1.push('0');
        x=0;
    }else if(a=='/' ){
        result=y/x;
        lista1.push('0');
        x=0;
    }
}
return result+"";
}

static LinkedList variables (String expresion){
    LinkedList<Character> lista=new LinkedList();
    int tam =expresion.length();
    for(int i=0;i<tam;i++){
        char a=expresion.charAt(i);
        if(Character.isLetter(a)){
            if(lista.contains(a)){
                lista.remove((Character)a);
            }else
                lista.add(a);
        }
    }
    return lista;
}
}

```