

CALCULADORA

Fernando Villalobos
Fernando Espinosa
Luis Eduardo Suarez
Oscar Martinez
Aldo Soria

Instituto Tecnológico Autónomo de México

21 de Septiembre de 2022

ÍNDICE

DESCRIPCIÓN DEL PROBLEMA	3
SOLUCIÓN DISEÑADA	4
PRUEBAS	5
LIMITACIONES DE LA SOLUCIÓN	6
POSIBLES MEJORAS Y CONCLUSIONES	7
APÉNDICE	8

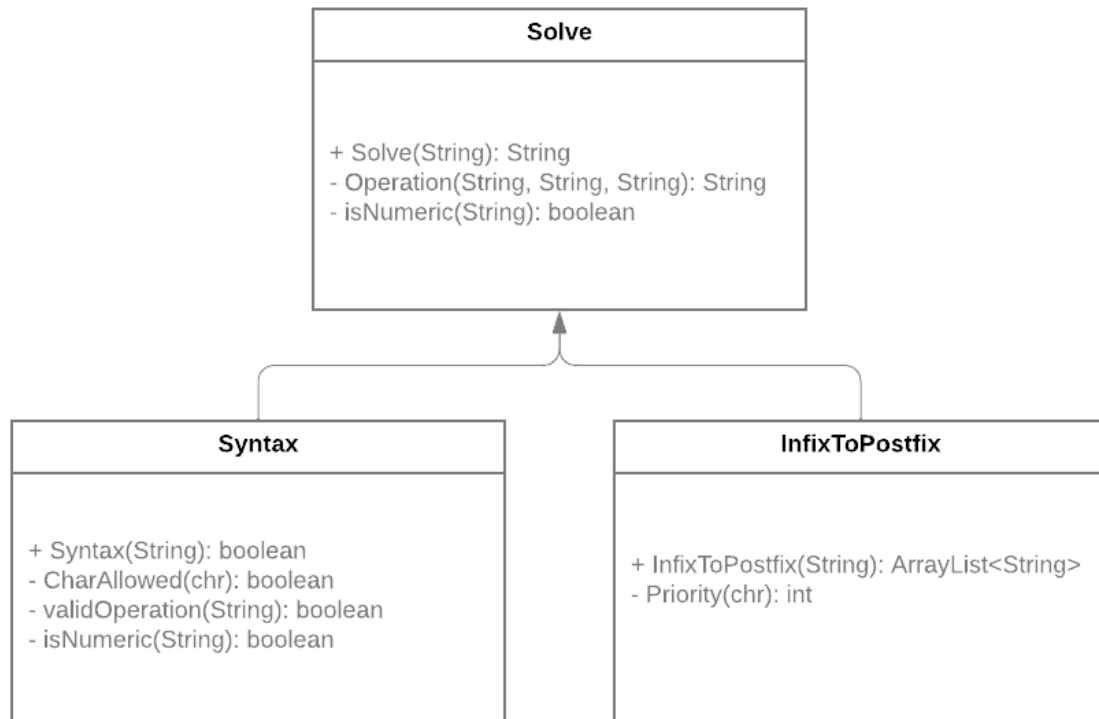
DESCRIPCIÓN DEL PROBLEMA

El proyecto está enfocado en el desarrollo de una aplicación que pueda usar procesos lógicos para facilitar la solución de problemas aritméticos de un nivel medio de complejidad. En otras palabras, un sistema que ayude a realizar operaciones matemáticas con un cierto grado de complejidad. El problema resulta en el diseño de una fórmula que pueda tomar una cadena de caracteres y la utilice para resolver la cadena de operaciones. Este método tiene varias complicaciones que se tienen que resolver para tener una calculadora funcional. Las dos más grandes complicaciones son: errores de escritura de la operación; y el traducir la cadena a un modo que la computadora pueda usar.

Primero, los errores de escritura en la operación se refiere a una expresión que utiliza mal los símbolos aritméticos y puedan provocar un error en el proceso de resolución. Por ello, es necesario crear un método que pueda detectar los errores y que invalide el método de resolución para evitar errores en la computadora. Segundo, traducir la cadena significa trasladar los caracteres del código a otra forma con la que la computadora pueda trabajar para dar solución a las expresiones. Este paso es importante ya que la calculadora usa pilas como forma de almacenamiento de datos y por la particularidad de ellas se requiere de diferentes métodos para poder usarlos en una calculadora. Esto requiere reorganizar los números y los símbolos para respetar las prioridades de operaciones aritméticas y no obtener resultados erróneos. Al final, el mayor reto de la calculadora es poder traducir para la computadora el orden de las funciones para seguir la prioridad de operaciones y el evitar errores que puedan tronar el código.

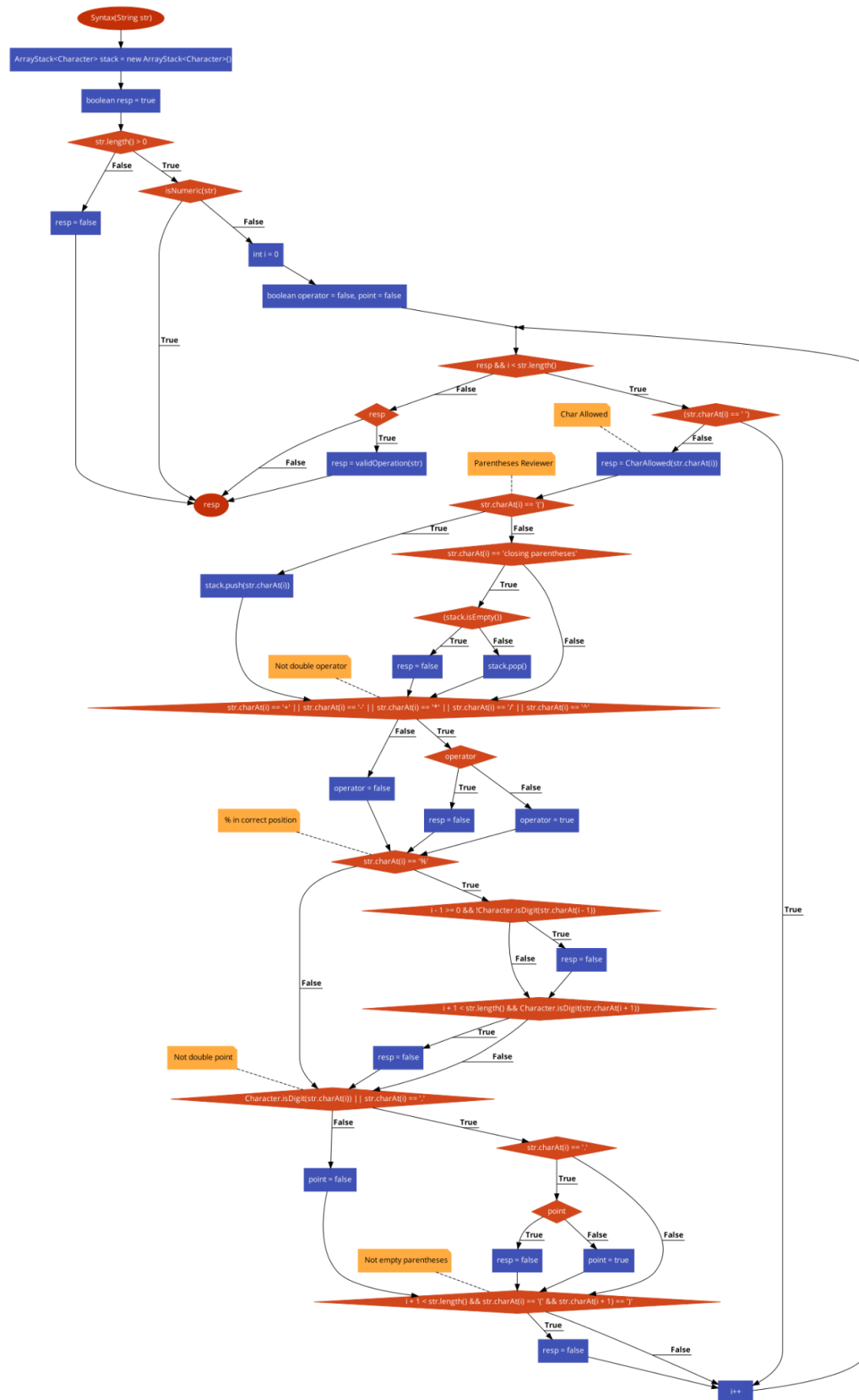
SOLUCIÓN DISEÑADA

UML



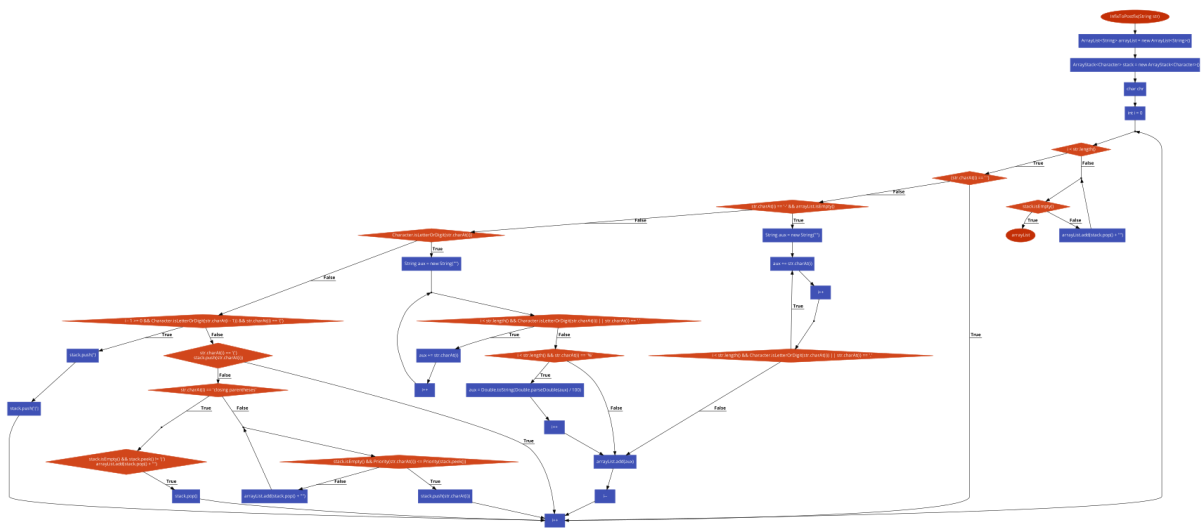
Algoritmos Principales

Syntax



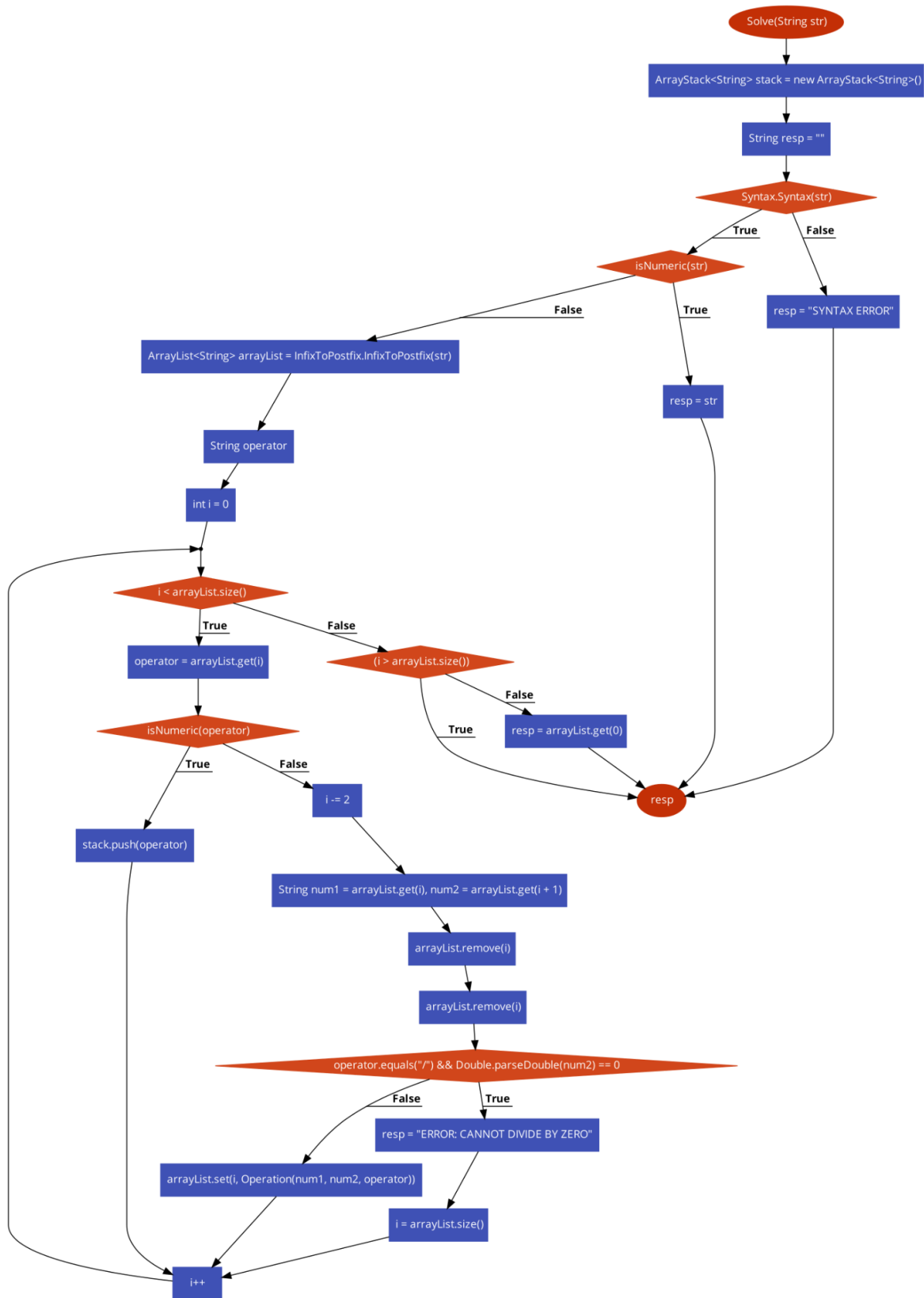
<https://app.code2flow.com/5aS3Vmuf1LXO.svg>

InfixToPostfix



<https://app.code2flow.com/qJ7u4x5Lh6NQ.svg>

Solve



<https://app.code2flow.com/ZgwTGHjubKnK.svg>

PRUEBAS

Las pruebas fueron realizadas en las 3 clases principales: Syntax, InfixToPostfix y Solve. Ya que estos son los algoritmos más complejos en el proyecto. Las pruebas realizadas fueron:

Syntax

- String vacío: "".
- Operación compleja con orden de operaciones: $3*((34+87)-3*(45)+(46/52)+(59+5))$.
- Paréntesis desbalanceados: $(95-8)*(89/4)$.
- Paréntesis desbalanceados en operación compleja: $3*(((648*5)-(978-65))+(1489/98))$.
- Caracteres no válidos: "Hola".
- Espacios en la operación: $3 * ((34 + 87) - 3(45) + (46 / 52) + (59 + 5))$
- Número sin operación: "3".
- Operación incompleta: "3+".
- Dobles operadores: $9-5*(3+-8)$.
- Operaciones con punto decimal: $3.8+8.5$.
- Operaciones con doble punto decimal: $33.8.5+8$.
- Porcentajes: $3\%*100$.
- Paréntesis como multiplicación: $3(7)$

InfixToPostfix

- String vacío: "".
- Operación compleja con orden de operaciones: $3*((34+87)-3(45)+(46/52)+(59+5))$.
- Paréntesis como multiplicación: $98+64(8489/45)$.
- Porcentajes: $25\%*100$.
- Operación con orden de operación: $354+891*18/878$.
- Potencias: 3541^5426 .

Solve

- De 1 a 100,000 operaciones random, sencillas: $23.156*4.654$, $48/486$, $1489-87$, $528+84$, 84^58 ..
- Operación compleja con orden de operaciones: $3*((34+87)-3(45)+(46/52)+(59+5))$.
- Paréntesis como multiplicación: $98+64(8489/45)$.
- Porcentajes: $25\%*100$.

LIMITACIONES DE LA SOLUCIÓN

La principal limitación que enfrentamos es que la calculadora aún tiene conflictos cuando se utiliza un número negativo (ej. -2) en cualquier tipo de operación.

Igualmente, otro aspecto a considerar es que al momento de hacer las pruebas notamos que los resultados que arroja nuestra calculadora difieren ligeramente con los resultados que arroja Java al momento de solucionar una ecuación. Sin embargo, nos dimos cuenta que nuestros resultados coinciden con los de otras calculadoras externas, a diferencia de los resultados que arroja Java.

POSIBLES MEJORAS Y CONCLUSIONES

En general, el código está bastante completo y es eficiente, sin embargo, eso no significa que no haya margen de mejora. Especialmente en torno a solucionar las limitaciones previamente enunciadas. Esto es la mejora o actualización del código para que pueda solucionar otro tipo de problemas matemáticos, como por ejemplo logaritmos y funciones trigonométricas tales como coseno, seno y tangente; también el agregar una funcionalidad que ayude al usuario a poner raíces de distintas potencias sin necesidad de ponerlo como una potencia en división. Además, agregando más funcionalidades al código, se puede programar para que este tenga una memoria, o sea, que pueda guardar el resultado obtenido, mejor conocido como “ans” en algunas otras calculadoras.

APÉNDICE

Syntax
Solve
InfixToPostfix
interface

```
package Syntax;

import Stack.ArrayStack;

public class Syntax {
    public static boolean Syntax(String str) {
        ArrayStack<Character> stack = new
ArrayStack<Character>();
        boolean resp = true;

        if (str.length() > 0) {
            if (!isNumeric(str)) {
                int i = 0;
                boolean operator = false, point = false;
```

```

while (resp && i < str.length()) {
    if (!(str.charAt(i) == ' ')) {

        // Char Allowed
        resp = CharAllowed(str.charAt(i));

        // Parentheses Reviewer
        if (Character.compare(str.charAt(i),
'(') == 0)
            stack.push(str.charAt(i));
        if (Character.compare(str.charAt(i),
')') == 0) {
            if (!(stack.isEmpty()))
                stack.pop();
            else resp = false;
        }

        // Not double operator
        if (str.charAt(i) == '+' || str.charAt(i)
== '-' || str.charAt(i) == '*' || str.charAt(i) == '/' ||
str.charAt(i) == '^') {
            if (operator)
                resp = false;
            else
                operator = true;
        } else
            operator = false;
    }
}

```

```

        // % in correct position
        if (str.charAt(i) == '%') {
            if (i - 1 >= 0 &&
!Character.isDigit(str.charAt(i - 1)))
                resp = false;
            if (i + 1 < str.length() &&
Character.isDigit(str.charAt(i + 1)))
                resp = false;
        }

        // Not double point
        if (Character.isDigit(str.charAt(i)) ||
str.charAt(i) == '.') {
            if (str.charAt(i) == '.') {
                if (point)
                    resp = false;
                else
                    point = true;
            }
        } else
            point = false;

        // Not empty parentheses
        if (i + 1 < str.length() &&
str.charAt(i) == '(' && str.charAt(i + 1) == ')')
            resp = false;
    }

    i++;

```

```

        }

        if (resp)
            resp = validOperation(str);
    }
    } else
        resp = false;

    return resp;
}

private static boolean CharAllowed(char chr) {

    return switch (chr) {
        case '(', ')', '.', '+', '-', '*', '/', '^', '%', '0', '1',
'2', '3', '4', '5', '6', '7', '8', '9' -> true;
        default -> false;
    };
}

private static boolean validOperation(String str)
{
    boolean num1 = false, operator = false,
num2 = false;
    int i = 0;

    while (!num2 && i < str.length()) {
        char chr = str.charAt(i);

```

```
        if (Character.isDigit(chr)) {
            if (num1 && operator)
                num2 = true;
            else
                num1 = true;
        } else {
            if (chr == '+' || chr == '-' || chr == '*' || chr
== '/' || chr == '^')
                operator = true;
        }

        i++;
    }

    return num1 && operator && num2;
}

private static boolean isNumeric(String
strNum) {
    if (strNum == null) {
        return false;
    }
    try {
        double d = Double.parseDouble(strNum);
    } catch (NumberFormatException nfe) {
        return false;
    }
    return true;
}
```

```
}
```

```
package Solve;
```

```
import InfixToPostfix.InfixToPostfix;
```

```
import Stack.ArrayStack;
```

```
import Syntax.Syntax;
```

```
import java.util.ArrayList;
```

```
public class Solve {
```

```
    public static String Solve(String str) {
```

```
        ArrayStack<String> stack = new
```

```
ArrayStack<String>();
```

```
        String resp = "";
```

```
        if (Syntax.Syntax(str)) {
```

```
            if (!isNumeric(str)) {
```

```
                ArrayList<String> arrayList =
```

```
InfixToPostfix.InfixToPostfix(str);
```

```
                String operator;
```

```
                int i = 0;
```

```
                while (i < arrayList.size()) {
```

```
                    operator = arrayList.get(i);
```

```
                    if (isNumeric(operator))
```

```
                        stack.push(operator);
```

```
                    else {
```

```
                        i -= 2;
```



```
        String num1 = arrayList.get(i),
num2 = arrayList.get(i + 1);

        arrayList.remove(i);
        arrayList.remove(i);

        if (operator.equals("/") &&
Double.parseDouble(num2) == 0) {
            resp = "ERROR: CANNOT
DIVIDE BY ZERO";
            i = arrayList.size();
        } else
            arrayList.set(i, Operation(num1,
num2, operator));
    }

    i++;
}

    if (!(i > arrayList.size()))
        resp = arrayList.get(0);
    } else
        resp = str;
} else
    resp = "SYNTAX ERROR";

return resp;
}
```

```
private static String Operation(String num1,
String num2, String operator) {
    double resp = 0;

    switch (operator) {
        case "+":
            resp = Double.parseDouble(num1) +
Double.parseDouble(num2);
            break;
        case "-":
            resp = Double.parseDouble(num1) -
Double.parseDouble(num2);
            break;
        case "*":
            resp = Double.parseDouble(num1) *
Double.parseDouble(num2);
            break;
        case "/":
            resp = Double.parseDouble(num1) /
Double.parseDouble(num2);
            break;
        case "^":
            resp =
Math.pow(Double.parseDouble(num1),
Double.parseDouble(num2));
            break;
    }
    return Double.toString(resp);
}
```

```
private static boolean isNumeric(String
strNum) {
    if (strNum == null) {
        return false;
    }
    try {
        double d = Double.parseDouble(strNum);
    } catch (NumberFormatException nfe) {
        return false;
    }
    return true;
}
}
```

```
public class InfixToPostfix {

    public static ArrayList<String>
    InfixToPostfix(String str) {
        ArrayList<String> arrayList = new
        ArrayList<String>();
        PilaA<Character> stack = new
        PilaA<Character>();
        char chr;
        int i = 0;

        while (i < str.length()) {
            if (!(str.charAt(i) == ' ')) {
```

```
        // If the first character is a minus so it is
        considered as part of the number and not an
        operator
```

```
        if (str.charAt(i) == '-' &&
        arrayList.isEmpty()) {
            String aux = new String("");
```

```
            aux += str.charAt(i);
            i++;
```

```
            while (i < str.length() &&
            (Character.isLetterOrDigit(str.charAt(i)) ||
            str.charAt(i) == '.')) {
                aux += str.charAt(i);
                i++;
            }
```

```
            arrayList.add(aux);
            i--;
        } // To keep numbers with decimals
        together
```

```
        else if (i + 1 < str.length() &&
        Character.isLetterOrDigit(str.charAt(i)) &&
        (Character.isLetterOrDigit(str.charAt(i + 1)) ||
        str.charAt(i + 1) == '.')) {
            String aux = new String("");
```

```
            while (i < str.length() &&
            (Character.isLetterOrDigit(str.charAt(i)) ||
```

```

str.charAt(i) == '.') {
    aux += str.charAt(i);
    i++;
}

    arrayList.add(aux);
    i--;
} // To consider parentheses also as a
multiplication operator
    else if (i - 1 >= 0 &&
Character.isLetterOrDigit(str.charAt(i - 1)) &&
str.charAt(i) == '(') {
    stack.push('*');
    stack.push('(');
} // To add simple numbers: 3, 4, 5
    else if
(Character.isLetterOrDigit(str.charAt(i))) {
    arrayList.add(str.charAt(i) + "");
} // To add a parentheses
    else if (str.charAt(i) == '(') {
    stack.push(str.charAt(i));
} // To acomodate all inside a
parentheses
    else if (str.charAt(i) == ')') {
    while (!stack.isEmpty() &&
stack.peek() != '(') {
        arrayList.add(stack.pop() + "");
    }
}

```

```

        stack.pop();
    } // To add all operators
    else {
        while (!stack.isEmpty() &&
Priority(str.charAt(i)) <= Priority(stack.peek())) {
            arrayList.add(stack.pop() + "");
        }

        stack.push(str.charAt(i));
    }
}

i++;
}

while (!stack.isEmpty()) {
    arrayList.add(stack.pop() + "");
}

return arrayList;
}

private static int Priority(char chr) {
    return switch (chr) {
        case '+', '-' ->
            1;
        case '*', '/' ->
            2;
        case '^' ->

```

```
        3;  
        default ->  
        -1;  
    };  
}  
}
```

```
public class Interface extends  
javax.swing.JFrame {
```

```
    public Interface() {  
        initComponents();  
    }
```

```
    @SuppressWarnings("unchecked")  
    // <editor-fold defaultstate="collapsed"  
    desc="Generated Code">  
    private void initComponents() {  
  
        jPanel1 = new javax.swing.JPanel();  
        jScrollPane1 = new  
javax.swing.JScrollPane();  
        jTextArea1 = new javax.swing.JTextArea();  
        jScrollPane2 = new  
javax.swing.JScrollPane();  
        jTextArea2 = new javax.swing.JTextArea();  
        jButton3 = new javax.swing.JButton();
```

```
jButton4 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();
jButton7 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
jButton8 = new javax.swing.JButton();
jButton9 = new javax.swing.JButton();
jButton10 = new javax.swing.JButton();
jButton11 = new javax.swing.JButton();
jButton12 = new javax.swing.JButton();
jButton13 = new javax.swing.JButton();
jButton14 = new javax.swing.JButton();
jButton15 = new javax.swing.JButton();
jButton16 = new javax.swing.JButton();
jButton17 = new javax.swing.JButton();
jButton18 = new javax.swing.JButton();
jButton19 = new javax.swing.JButton();
jButton20 = new javax.swing.JButton();
jButton21 = new javax.swing.JButton();
jButton22 = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowC
onstants.EXIT_ON_CLOSE);
```

```
jTextArea1.setColumns(20);
jTextArea1.setRows(5);
jTextArea1.setText("operacion a resolver");
jScrollPane1.setViewportView(jTextArea1);
```



```
jTextArea2.setColumns(20);
jTextArea2.setRows(5);
jTextArea2.setText("resultado");
jScrollPane2.setViewportViewView(jTextArea2);

jButton3.setText("(");
jButton3.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt)
{
    jButton3ActionPerformed(evt);
}
});

jButton4.setText("^");
jButton4.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt)
{
    jButton4ActionPerformed(evt);
}
});

jButton6.setText(")");
jButton6.addActionListener(new
java.awt.event.ActionListener() {
    public void
```

```
actionPerformed(java.awt.event.ActionEvent evt)
{
    JButton6ActionPerformed(evt);
}

});

JButton7.setText("/");
JButton7.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt)
{
    JButton7ActionPerformed(evt);
}

});

JButton5.setText("9");
JButton5.addActionListener(new
java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.ActionEvent evt)
{
    JButton5ActionPerformed(evt);
}

});

JButton8.setText("8");
JButton8.addActionListener(new
java.awt.event.ActionListener() {
```

```
        public void
actionPerformed(java.awt.event.ActionEvent evt)
{
    JButton8ActionPerformed(evt);
}
});

    JButton9.setText("");
    JButton9.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt)
{
            JButton9ActionPerformed(evt);
        }
    });

    JButton10.setText("7");
    JButton10.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt)
{
            JButton10ActionPerformed(evt);
        }
    });

    JButton11.setText("6");
    JButton11.addActionListener(new
```

```
java.awt.event.ActionListener() {  
    public void  
    actionPerformed(java.awt.event.ActionEvent evt)  
{  
    JButton11ActionPerformed(evt);  
    }  
});  
  
JButton12.setText("5");  
JButton12.addActionListener(new  
java.awt.event.ActionListener() {  
    public void  
    actionPerformed(java.awt.event.ActionEvent evt)  
{  
    JButton12ActionPerformed(evt);  
    }  
});  
  
JButton13.setText("+");  
JButton13.addActionListener(new  
java.awt.event.ActionListener() {  
    public void  
    actionPerformed(java.awt.event.ActionEvent evt)  
{  
    JButton13ActionPerformed(evt);  
    }  
});  
  
JButton14.setText("4");
```

```
        jButton14.addActionListener(new  
java.awt.event.ActionListener() {  
            public void  
actionPerformed(java.awt.event.ActionEvent evt)  
{  
                jButton14ActionPerformed(evt);  
            }  
        });
```

```
        jButton15.setText("3");  
        jButton15.addActionListener(new  
java.awt.event.ActionListener() {  
            public void  
actionPerformed(java.awt.event.ActionEvent evt)  
{  
                jButton15ActionPerformed(evt);  
            }  
        });
```

```
        jButton16.setText("2");  
        jButton16.addActionListener(new  
java.awt.event.ActionListener() {  
            public void  
actionPerformed(java.awt.event.ActionEvent evt)  
{  
                jButton16ActionPerformed(evt);  
            }  
        });
```

```
        jButton17.setText("-");
        jButton17.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt)
{
                jButton17ActionPerformed(evt);
            }
});
```

```
        jButton18.setText("1");
        jButton18.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt)
{
                jButton18ActionPerformed(evt);
            }
});
```

```
        jButton19.setText(".");
        jButton19.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt)
{
                jButton19ActionPerformed(evt);
            }
});
```

```
        jButton20.setText("0");
        jButton20.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt)
{
                jButton20ActionPerformed(evt);
            }
});
```

```
        jButton21.setText("=");
        jButton21.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt)
{
                jButton21ActionPerformed(evt);
            }
});
```

```
        jButton22.setText("Clear");
        jButton22.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt)
{
                jButton22ActionPerformed(evt);
            }
});
```

```
});
```

```
    javax.swing.GroupLayout jPanel1Layout =  
new javax.swing.GroupLayout(jPanel1);  
    jPanel1.setLayout(jPanel1Layout);  
    jPanel1Layout.setHorizontalGroup(  

```

```
jPanel1Layout.createParallelGroup(javax.swing.  
GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel1Layout.createSequentialGroup  
(
```

```
.addGroup(jPanel1Layout.createParallelGroup(ja  
vax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel1Layout.createSequentialGroup  
(
```

```
        .addComponent(jScrollPane1)  
        .addGap(18, 18, 18)  
        .addComponent(jScrollPane2))
```

```
.addGroup(jPanel1Layout.createSequentialGroup  
(
```

```
        .addContainerGap()
```

```
.addGroup(jPanel1Layout.createParallelGroup(ja  
vax.swing.GroupLayout.Alignment.LEADING)
```



```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addComponent(jButton14,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
    .addGap(18, 18, 18)
    .addComponent(jButton12,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
    .addGap(18, 18, 18)
    .addComponent(jButton11,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
    .addGap(18, 18, 18)
    .addComponent(jButton13,
javax.swing.GroupLayout.DEFAULT_SIZE, 160,
Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup
())
    .addComponent(jButton18,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
    .addGap(18, 18, 18)
    .addComponent(jButton16,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
    .addGap(18, 18, 18)
```

```
        .addComponent(jButton15,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(jButton17,
javax.swing.GroupLayout.DEFAULT_SIZE, 160,
Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup
())

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton10,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
        .addComponent(jButton3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton6,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jButton8,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE))
        .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jButton5,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE))
        .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton7,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jButton9,
javax.swing.GroupLayout.DEFAULT_SIZE, 160,
Short.MAX_VALUE))))))

.addGroup(javax.swing.GroupLayout.Alignment.T
RAILING,
jPanel1Layout.createSequentialGroup()
        .addGap(6, 6, 6)
```

```
        .addComponent(jButton22,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(jButton20,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(jButton19,
javax.swing.GroupLayout.DEFAULT_SIZE, 159,
Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(jButton21,
javax.swing.GroupLayout.DEFAULT_SIZE, 160,
Short.MAX_VALUE)))
        .addContainerGap()
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup
())
        .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.LEADING,
false)
```

```
.addComponent(jScrollPane2)
.addComponent(jScrollPane1))
.addGap(18, 18, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.LEADING,
false)

.addComponent(jButton4,
javax.swing.GroupLayout.DEFAULT_SIZE, 71,
Short.MAX_VALUE)
.addComponent(jButton6,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
.addComponent(jButton3,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
.addComponent(jButton7,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(ja
vax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton5,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
        .addComponent(jButton9,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
        .addComponent(jButton8,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
        .addComponent(jButton10,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE))  
        .addGap(18, 18, 18)  
  
        .addGroup(jPanel1Layout.createParallelGroup(ja  
vax.swing.GroupLayout.Alignment.BASELINE)  
        .addComponent(jButton11,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
        .addComponent(jButton13,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
        .addComponent(jButton12,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
        .addComponent(jButton14,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE))  
        .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(ja  
vax.swing.GroupLayout.Alignment.BASELINE)  
    .addComponent(jButton15,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
    .addComponent(jButton17,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
    .addComponent(jButton16,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
    .addComponent(jButton18,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE))  
    .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(ja  
vax.swing.GroupLayout.Alignment.BASELINE)  
    .addComponent(jButton19,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
    .addComponent(jButton21,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
    .addComponent(jButton20,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,  
Short.MAX_VALUE)  
    .addComponent(jButton22,  
javax.swing.GroupLayout.DEFAULT_SIZE, 63,
```

```
Short.MAX_VALUE))
        .addGap(60, 60, 60))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout
out.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout
out.Alignment.LEADING)
        .addComponent(jPanel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );
```



```
    pack();  
} // </editor-fold>
```

```
StringBuilder sb = new StringBuilder();
```

```
    private void  
jButton3ActionPerformed(java.awt.event.ActionEvent  
    evt) {  
        sb.append("(");  
        jTextArea1.setText("");  
        jTextArea1.append(sb.toString());  
  
    }
```

```
    private void  
jButton4ActionPerformed(java.awt.event.ActionEvent  
    evt) {  
        // TODO add your handling code here:  
        sb.append("^");  
        jTextArea1.setText("");  
        jTextArea1.append(sb.toString());  
  
    }
```

```
    private void  
jButton6ActionPerformed(java.awt.event.ActionEvent  
    evt) {  
        // TODO add your handling code here:
```

```
        sb.append("");
        jTextArea1.setText("");
        jTextArea1.append(sb.toString());
    }
```

```
    private void
jButton7ActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        sb.append("/");
        jTextArea1.setText("");
        jTextArea1.append(sb.toString());
    }
```

```
    private void
jButton5ActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
        sb.append("9");
        jTextArea1.setText("");
        jTextArea1.append(sb.toString());

    }
```

```
    private void
jButton8ActionPerformed(java.awt.event.ActionEvent
    evt) {
        // TODO add your handling code here:
```

```
        sb.append("8");
        jTextArea1.setText("");
        jTextArea1.append(sb.toString());
    }
```

```
    private void
jButton9ActionPerformed(java.awt.event.ActionE
vent evt) {
    // TODO add your handling code here:
    sb.append("*");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
}
```

```
    private void
jButton10ActionPerformed(java.awt.event.Action
Event evt) {
    sb.append("7");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
    // TODO add your handling code here:
}
```

```
    private void
jButton11ActionPerformed(java.awt.event.Action
Event evt) {
    // TODO add your handling code here:
    sb.append("6");
    jTextArea1.setText("");
}
```

```
        jTextArea1.append(sb.toString());
    }

    private void
jButton12ActionPerformed(java.awt.event.Action
Event evt) {
    // TODO add your handling code here:
    sb.append("5");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
}

    private void
jButton13ActionPerformed(java.awt.event.Action
Event evt) {
    // TODO add your handling code here:
    sb.append("+");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
}

    private void
jButton14ActionPerformed(java.awt.event.Action
Event evt) {
    sb.append("4");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());// TODO
add your handling code here:
}
```

```
private void
jButton15ActionPerformed(java.awt.event.ActionEvent evt) {
    sb.append("3");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString()); // TODO
add your handling code here:
}
```

```
private void
jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
    sb.append("2");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
// TODO add your handling code here:
}
```

```
private void
jButton17ActionPerformed(java.awt.event.ActionEvent evt) {
    sb.append("-");
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
// TODO add your handling code here:
}
```

```
private void
```

```
jButton18ActionPerformed(java.awt.event.ActionEvent evt) {  
    sb.append("1");  
    jTextArea1.setText("");  
    jTextArea1.append(sb.toString()); // TODO  
    add your handling code here:  
}
```

```
private void  
jButton19ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    sb.append(".");  
    jTextArea1.setText("");  
    jTextArea1.append(sb.toString());  
}
```

```
private void  
jButton20ActionPerformed(java.awt.event.ActionEvent evt) {  
    // inserta el numero 0  
    sb.append("0");  
    jTextArea1.setText("");  
    jTextArea1.append(sb.toString());  
}
```

```
private void  
jButton21ActionPerformed(java.awt.event.ActionEvent evt) {
```

```

// TODO add your handling code here:
//este es el boton igual
StringBuilder sb2 = new StringBuilder();
sb2.append(Solve(sb.toString()));
jTextArea2.setText("");
jTextArea2.setText(sb2.toString());
sb= new StringBuilder();
jTextArea1.setText("");
jTextArea1.append(sb.toString());
}

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sb= new StringBuilder();
    jTextArea1.setText("");
    jTextArea1.append(sb.toString());
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed"
    desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not
    available, stay with the default look and feel.

```

```

    * For details see
    http://download.oracle.com/javase/tutorial/uiswing
    /lookandfeel/plaf.html
    */
    try {
        for
        (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeel
        s()) {
            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.get
                ClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Interface.class
        .getName()).log(java.util.logging.Level.SEVERE,
        null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Interface.class
        .getName()).log(java.util.logging.Level.SEVERE,
        null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Interface.class

```



```
.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
    } catch  
(javax.swing.UnsupportedLookAndFeelException  
ex) {
```

```
java.util.logging.Logger.getLogger(Interface.class  
.getName()).log(java.util.logging.Level.SEVERE,  
null, ex);  
    }  
    //</editor-fold>
```

```
    /* Create and display the form */  
    java.awt.EventQueue.invokeLater(new  
Runnable() {  
        public void run() {  
            new Interface().setVisible(true);  
        }  
    });  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JButton jButton10;  
private javax.swing.JButton jButton11;  
private javax.swing.JButton jButton12;  
private javax.swing.JButton jButton13;  
private javax.swing.JButton jButton14;  
private javax.swing.JButton jButton15;  
private javax.swing.JButton jButton16;
```

```
private javax.swing.JButton jButton17;  
private javax.swing.JButton jButton18;  
private javax.swing.JButton jButton19;  
private javax.swing.JButton jButton20;  
private javax.swing.JButton jButton21;  
private javax.swing.JButton jButton22;  
private javax.swing.JButton jButton3;  
private javax.swing.JButton jButton4;  
private javax.swing.JButton jButton5;  
private javax.swing.JButton jButton6;  
private javax.swing.JButton jButton7;  
private javax.swing.JButton jButton8;  
private javax.swing.JButton jButton9;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane2;  
private javax.swing.JTextArea jTextArea1;  
private javax.swing.JTextArea jTextArea2;  
// End of variables declaration  
}
```