UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

Nombre del curso:

ESTRUCTURA DE DATOS

Profesor:

Jesús Hernández Cabrera

Tarea:

Tarea 9.- Evaluación de balanceo de paréntesis y llaves

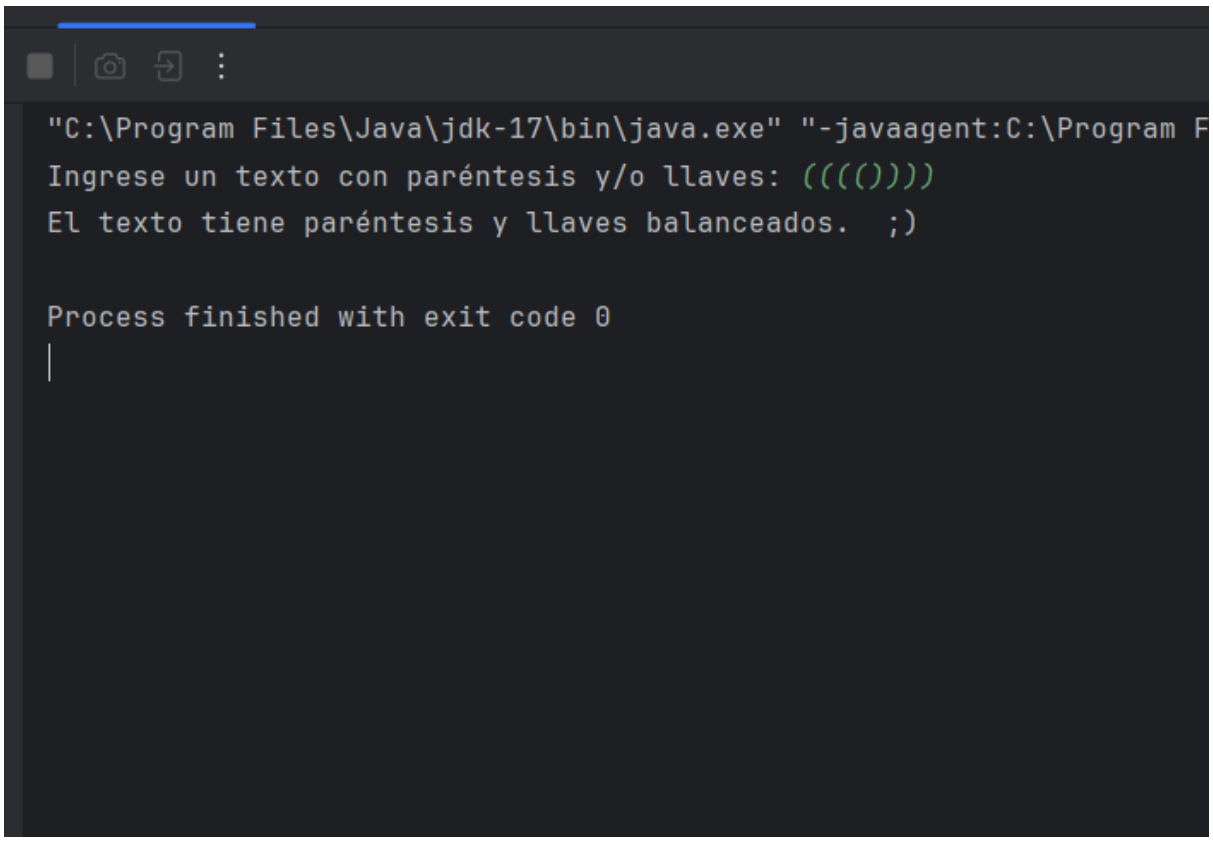
Alumno:

Navarro Rocha Miguel Ángel

Fecha:

25-09-24

Funcionamiento:



Código MAIN:

```
package Tarea9;

import java.util.Scanner;

public class Main {

    // función que verifica si los paréntesis y llaves están balanceados
    public static boolean areParenthesesBalanced(String expr) {
        Stack<Character> stack = new Stack<>();

        // para recorrer cada carácter en la expresión
        for (int i = 0; i < expr.length(); i++) {
            char currentChar = expr.charAt(i);

            // si el carácter es una llave o paréntesis de apertura, lo añade a la pila
            if (currentChar == '(' || currentChar == '{') {
                stack.push(currentChar);
            }
            // si es una llave o paréntesis de cierre, verifica el balanceo
            else if (currentChar == ')' || currentChar == '}') {
                // si la pila está vacía, hay un error (falta una llave de apertura )
                if (stack.isEmpty()) {
                    return false;
                }
                char topChar = stack.pop();

                // verifica si los caracteres coinciden como apertura y cierre
                if ((currentChar == ')' && topChar != '(') || (currentChar == '}' && topChar != '{')) {
                    return false; // no coinciden
                }
            }
        }

        // si la pila está vacía al final, significa que esta balanceado
        return stack.isEmpty();
    }

    public static void main(String[] args) {
        // crear un objeto Scanner para capturar la entrada del usuario
        Scanner scanner = new Scanner(System.in);

        // pide al usuario que ingrese una cadena de texto
        System.out.print("Ingrese un texto con paréntesis y/o llaves: ");
        String inputText = scanner.nextLine(); // Capturar la cadena de entrada

        // verificar si la entrada del usuario está balanceada
        boolean resultado = areParenthesesBalanced(inputText);

        // mostrar el resultado
        if (resultado) {
            System.out.println("El texto tiene paréntesis y llaves balanceados.  ;) ");
        } else {
            System.out.println("El texto NO tiene paréntesis y/o llaves balanceados.  :( ");
        }

        // cierra el escáner
        scanner.close();
    }
```

```
}
```

Código Stack:

```
package Tarea9;

import java.util.ArrayList;

class Stack<T> {
    private ArrayList<T> data;

    public Stack() {
        this.data = new ArrayList<>();
    }

    // pila vacía
    public boolean isEmpty() {
        return data.isEmpty();
    }

    // devuelve la longitud de pila
    public int length() {
        return data.size();
    }

    // extrae el último elemento de la pila (LIFO)
    public T pop() {
        if (!isEmpty()) {
            return data.remove(data.size() - 1);
        }
        return null; // error si está vacía
    }

    // devuelve el último elemento sin extraerlo
    public T peek() {
        if (!isEmpty()) {
            return data.get(data.size() - 1);
        }
        return null;
    }

    // añade un elemento a la pila
    public void push(T value) {
        data.add(value);
    }

    // representación en cadena de la pila
    @Override
    public String toString() {
        StringBuilder info = new StringBuilder("-----\n");
        for (int i = data.size() - 1; i >= 0; i--) {
            info.append(data.get(i)).append("\n|---|\n");
        }
        return info.toString();
    }
}
```