

ANDY ROBERTO JIMENEZ MACNAB

202111490

Laboratorio de Lenguajes Formales y de Programación

Sección B+

Manual Técnico

Analizador Léxico

25/03/2023

Universidad de San Carlos de Guatemala

Proyecto 1

Principio de Programación:

- Programación orientada a objetos:

En este modelo de paradigma se construyen modelos de objetos

que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.

Convenciones de Nomenclatura:

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: MiBlogDeDesarrollo

## convertToTextPlane():

Este método se encarga de convertir el archivo JSON ingresado por el usuario y posteriormente cargado en el textArea de tkinter en un texto plano para mayor facilidad para el programa al momento de analizarlo.

Este código implementa un analizador léxico y un parser básico para interpretar expresiones aritméticas y trigonométricas simples. La entrada es una cadena que representa una secuencia de instrucciones matemáticas separadas por líneas. La cadena se descompone en lexemas (tokens) que se clasifican en diferentes tipos como números, operadores aritméticos y trigonométricos, y paréntesis. La función "Instruccion" recibe la cadena de entrada y devuelve una lista de objetos "Lexema" que representan cada token en la secuencia. Luego, la función "operar\_" utiliza la lista de lexemas para evaluar la secuencia de instrucciones y devolver una lista de objetos "Aritmetica" o "Trigonometricas" que representan las operaciones aritméticas o trigonométricas a realizar

```
def GraficandoDatos(texto, colorFondo, colorFuente, forma, nodos, respuestas):
    spanishColors={
        "Negro":"black", "Rojo":"red", "Amarillo":"yellow", "Verde":"green", "Naranja":"orange", "Morado":"purple", "Blanco":"white"
        , "Cafe":"brown", "Azul":"blue", "Gris":"gray", "Rosado":"pink", "Cian":"cyan", "Corinto":"maroon"
    }
    spanishFigures={
        "Cuadrado":"square", "Circulo":"circle", "Triangulo":"triangle", "Trapezoide":"trapezoid", "Estrella":"star", "Rectangulo":"rectangle"
        , "Octagono":"octagon", "Corazon":"heart", "Diamante":"diamond", "Rombo":"rhombus", "Hexagono":"hexagon", "Elipse":"ellipse"
    }
}
```

Este código es una función llamada "GraficandoDatos" que se encarga de crear un diagrama utilizando la biblioteca Graphviz en Python. La función toma como entrada varias variables, como el texto que se utilizará para el título del diagrama, el color de fondo, el color de la fuente, la forma de los nodos, una lista de nodos y una lista de respuestas.

Dentro de la función, se crea un objeto Digraph de Graphviz y se define un subgrafo con el título y los estilos definidos por las variables de entrada. Luego, se utiliza un ciclo while para iterar sobre la lista de nodos, en la que se encuentran las palabras clave "Operacion", "Valor1" y "Valor2".

Cuando se encuentra la palabra clave "Operacion", se verifica si se ha procesado previamente un valor anidado o si es el primer valor. Luego, se crea una cadena de caracteres que se agrega a la lista de nodos y se incrementa el puntero. Finalmente, se conecta el nodo actual con el nodo anterior utilizando el método "edge".