Universidad San Carlos De Guatemala

Centro Universitario De Occidente

División De Ciencias De La Ingeniería

Lenguajes Formales y de Programación

Auxiliar Julio Fernando

Manual Técnico

202132299, Gérman Alex Ramírez Limatuj

Quetzaltenango, Agosto de 2023

Manual técnico

Analizador Léxico

Parse Py

Parse-py (parte del Analizador Léxico)

Este manual técnico proporciona una guía detallada sobre el diseño y la funcionalidad del Analizador Léxico Interactivo en Java. El objetivo de este proyecto es permitir a los usuarios analizar el código fuente en un lenguaje similar a Python, identificar y visualizar los tokens y errores léxicos, y generar gráficos para representar mejor la estructura del código.

Entorno de Desarrollo

Para el desarrollo de esta aplicación se utilizó lo siguiente:

- JDK 17

- IDE Apache Netbeans

- Sistema Operativo Debian GNU/Linux 12 (bookworm) x86_64

- diagrams.net para la realización de diagrama de clases

Estructura del Proyecto

Paquete BackEnd.Herramientas

En este paquete se encuentran las clases que brindan funcionalidades auxiliares para el análisis léxico.

Clase Token

La clase Token almacena información sobre los tokens identificados en el código fuente. Cada token tiene un tipo, un lexema, un valor y coordenadas de fila y columna.

```
public class Token {
    // Atributos y constructor
    // Métodos de acceso
}
```

Clase Lexer

La clase Lexer se encarga del análisis léxico del código fuente y genera una lista de tokens. Utiliza expresiones regulares para identificar patrones en el texto.

```
public class Lexer {
    public void analyze(String sourceCode) {
        // Implementación del análisis léxico
    }

    public List<Token> getTokens() {
        // Obtener lista de tokens
```

```
    }
}
```

Clase OpenFile

La clase OpenFile permite cargar un archivo de texto en un componente de edición.

```
public class OpenFile {
    public static void openFileAndSetText(JTextPane textPane) {
        // Implementación para abrir y cargar archivo en el textPane
    }

    public static void loadTxtFile(File file, JTextPane textPane) {
        // Cargar contenido del archivo en el textPane
    }
}
```

Paquete FrontEnd.Graphviz

Este paquete contiene clases para generar y visualizar gráficos utilizando la herramienta Graphviz.

Clase Graphviz

La clase Graphviz genera gráficos a partir de código DOT y los guarda como imágenes.

```
public class Graphviz {
    public static void generateGraphic(String dotCode, String outputFileName) {
        // Generar archivo DOT temporal y ejecutar Graphviz
    }
}
```

Clase ViewGraphic

La clase ViewGraphic muestra gráficos generados en una interfaz gráfica.

```
public class ViewGraphic {
    public void displayGraph(String dotCode) {
        // Mostrar gráfico generado en una interfaz gráfica
    }
}
```

Paquete FrontEnd

En este paquete se encuentran las clases relacionadas con la interfaz gráfica y la ejecución del programa.

Clase EntornoVisual

La clase EntornoVisual representa la interfaz gráfica principal de la aplicación.

```
public class EntornoVisual extends JFrame {
    // Atributos y componentes
    public EntornoVisual() {
        // Configuración de la interfaz gráfica
    }

    private void analyzeCode() {
        // Realizar análisis léxico y mostrar resultados
    }

    private void generateGraph() {
        // Generar gráfico y mostrarlo en la interfaz
    }

    // Otros métodos y manejo de eventos
}
```

Clase ColoreoTokens

La clase ColoreoTokens se encarga de colorear los tokens en el JTextPane.

```
public class ColoreoTokens {
    public ColoreoTokens(JTextPane textPane) {
        // Constructor
    }

    public void colorearTokens(List<Token> tokens) {
        // Colorear tokens en el JTextPane
    }

    private Color getColorForTokenType(String tokenType) {
        // Obtener color correspondiente al tipo de token
```

```
    }
}
```

Uso de Swing para la Interfaz Gráfica

La biblioteca Swing se utiliza para crear la interfaz gráfica de la aplicación. Aquí se muestra cómo se utilizan algunos componentes de Swing en la clase EntornoVisual.

```
public class EntornoVisual extends JFrame {
    // ...

    private void initComponents() {
        JTextPane codeTextPane = new JTextPane();
        JScrollPane codeScrollPane = new JScrollPane(codeTextPane);

        JButton analyzeButton = new JButton("Analizar");
        analyzeButton.addActionListener(e -> analyzeCode());

        JButton generateGraphButton = new JButton("Generar Gráfico");
        generateGraphButton.addActionListener(e -> generateGraph());

        JTable tokenTable = new JTable();
        DefaultTableModel tableModel = new DefaultTableModel();
        tokenTable.setModel(tableModel);

        // Configuración del diseño de la interfaz gráfica
        // Agregar componentes a paneles y frames
    }

    // ...
}
```

Clase Main

La clase Main es el punto de entrada de la aplicación. Desde aquí se crea y muestra la interfaz gráfica principal EntornoVisual.

```
package BackEnd;


import FrontEnd.EntornoVisual;


/**
 * Punto de entrada de la aplicación.
 */
public class Main {
    public static void main(String[] args) {
        // Crear una instancia de EntornoVisual
        EntornoVisual entorno = new EntornoVisual();


        // Configurar la ubicación y visibilidad del entorno
        entorno.setLocationRelativeTo(null); // Centrar la ventana en la pantalla
        entorno.setVisible(true); // Mostrar la ventana
    }
}
```

La clase Main crea una instancia de EntornoVisual, que es la interfaz gráfica principal de la aplicación. Configura su ubicación para que se centre en la pantalla y la hace visible, lo que permite a los usuarios interactuar con la aplicación. La ejecución del método main inicia la aplicación y la hace funcional para el usuario.

Clase NumeracionFilas

La clase NumeracionFilas proporciona una numeración automática en las filas de un componente de texto, como JTextPane o JTextArea. Además, se encarga de actualizar la numeración en función del desplazamiento del cursor.

Esta clase es útil para mejorar la experiencia del usuario al mostrar números de línea, lo que facilita la referencia a una línea específica en un texto largo. El código se encarga de calcular y mostrar la numeración de las filas, centrándola verticalmente en cada línea del componente de texto.

Clase NumeracionColumnas

La clase NumeracionColumnas es responsable de generar y mostrar la numeración de columnas en un JTextPane, además de indicar la columna actual en la que se encuentra el cursor.

Esta clase mejora la interacción del usuario al proporcionar información sobre la posición horizontal exacta del cursor dentro del texto. Al detectar los cambios en el cursor, actualiza la numeración de columnas en función de la posición actual del cursor en la línea.