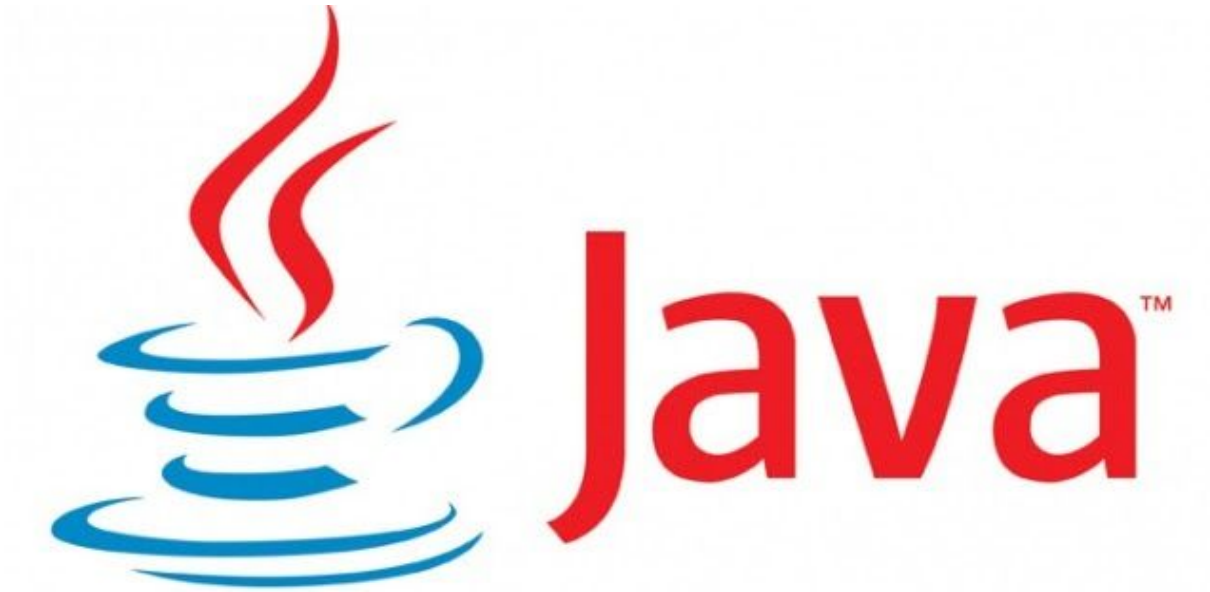
Manual Técnico

Entorno de desarrollo

Lenguaje de desarrollo: java en su versión SE development kit 11



El ide usado para programar es Netbeans en su versión 11

Diagrama de clases

Expression

Está clase sirve para guardar Objetos que almacenaran los siguientes campos:

- String Lexical_component: este atributo guardará el componente léxico el cual servirá para saber poder mandar a llamar un tipo de expresión regular
- String Lexeme: dicho atributo almacena el atributo lexema que es la entrada a analizar

Init

Es la clase principal está cuenta con el método main, en esta página creamos un objeto de tipo ventana llamado panel, que es el que nos proporciona la interfaz gráfica, contamos con una variable pública y estática que es de tipo String la cual se llama folder y sirve para almacenar el path absoluto de la carpeta donde serán almacenados los archivos tipo dot generados para poder crear las imágenes respectivas de las expresiones regulares

Está clase cuenta con un método que es encargado de crear una carpeta donde se encuentra el proyecto y guardar el path absoluto de la misma.

Panel_Init

Es la clase que nos proporciona la interfaz gráfica y posee métodos que realizan el análisis léxico de JTexArea definido como input_file, las variables definidas son las siguientes:

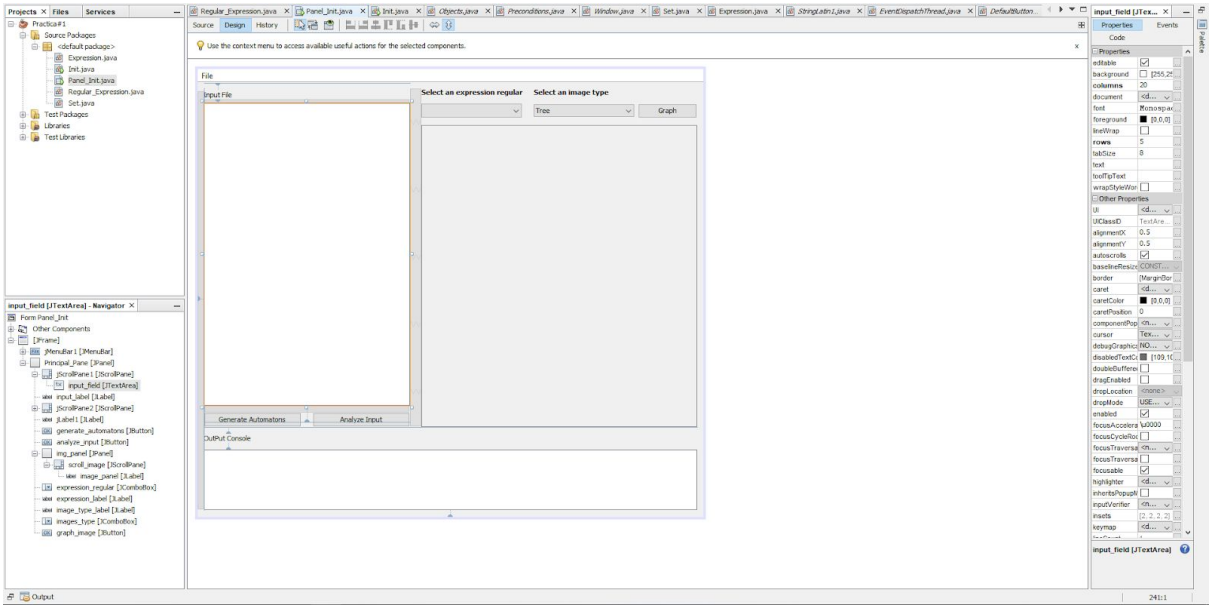
- private String path_file: almacena la ruta absoluta del archivo seleccionado para abrir, esto para luego guardar los cambios que realizo el usuario al campo de entrada.
- private ArrayList<Regular_Expression> regular_expre: es un arreglo el cual almacenara objetos de tipo expresión regular.
- private ArrayList<Set> sets; es un arreglo que guarda los objetos de tipo conjunto que fueron definidos por la entrada del usuario.
- private ArrayList<Expression> lexemes: Este arreglo almacena Objetos tipo lexema que hayan sido definidos por la entrada dada por el usuario.

Los métodos que podemos encontrar en está clase serán los siguientes:

- OpenAction
- SaveAction
- SaveAs
- GenerateAutomatons: Este método es el encargado de crear los objetos tipo Expresión Regular recolectando sus atributos del análisis que se realiza a la entrada proporcionada por el usuario.

- GraphImage: Este método obtiene los items seleccionados de los Jcombobox que definiran la imagen a crear y visualizar
- AnalizeInput: Es el método que realiza las búsquedas de los lexemas que serán empatados con el patrón de una expresión regular.
- AnalizeExpression: Es el encargado de verificar que la entrada buscar la expresión regular que será usada para analizar el lexema.
- GetExpresion
- OpenFIle: Este método crea un objeto de tipo JFilechooser el cual crea una ventana para obtener la ruta del archivo que quiere abrir el usuario para luego enviar el contenido al JTextArea.
- SaveFile:Obtiene el texto almacenado en el JtextArea en el archivo actual

El modo diseño de la interfaz gráfica el siguiente el cual muestra el nombre de los componentes usados como lo son los botones panels entre otros.



la definición de los componentes usados para esta clase es la siguiente

```
Principal_Pane = new javax.swing.JPanel();
jScrollPanel = new javax.swing.JScrollPane();
input_field = new javax.swing.JTextArea();
input_label = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
console = new javax.swing.JTextArea();
jLabel1 = new javax.swing.JLabel();
generate_automatons = new javax.swing.JButton();
analyze_input = new javax.swing.JButton();
img_panel = new javax.swing.JPanel();
scroll_image = new javax.swing.JScrollPane();
image_panel = new javax.swing.JLabel();
expression_regular = new javax.swing.JComboBox<>();
```

```
expression_label = new javax.swing.JLabel();
image_type_label = new javax.swing.JLabel();
images_type = new javax.swing.JComboBox<>();
graph_image = new javax.swing.JButton();
jMenuBar1 = new javax.swing.JMenuBar();
file_menu = new javax.swing.JMenu();
open = new javax.swing.JMenuItem();
save = new javax.swing.JMenuItem();
save_as = new javax.swing.JMenuItem();
generate_XML = new javax.swing.JMenuItem();
```

Regular Expression

Está clase define a los Objetos de tipo Expresión Regular los siguientes atributos:

- String pattern: Este atributo almacena el patrón de la expresión regular.
- String lexical_component: Almacena el nombre referente al lexema-
- Stack<Node> elements, operators;: Son dos pilas las cuales se usan para poder crear el árbol que define a la respectiva expresión regular
- int num_elements = 0: variable que lleva el control de cuantos elementos se elementos se encuentran en las pilas
- Node root: almacena la estructura del árbol de la expresión regular.
- ArrayList<Leaf> follows: arreglo de objetos tipo hoja.
- ArrayList<String> nonterminals: Guarda todos los no terminales válidos para la expresión regular.
- ArrayList<Terminal> terminals. Guarda los terminales que se crean del autómata.
- ArrayList<Transition> transitions. ALmacena objetos de tipo Transición.

Los métodos o funciones de la clase son los siguientes:

```
public Regular_Expression(String pattern, String lexical_comp)
{...11 lines }

public void generate_tree_expression()
{...58 lines }

public void insert_Nodes()
{...27 lines }

private void assing_number_leaf()
{...11 lines }

private void assing_number_leaf(Node pivot)
{...16 lines }

private void define_defeasible()
{...3 lines }
```

```
private void define_defeasible()
{...3 lines }

private void define_defeasible(Node pivot)
{...134 lines }

private void define_follows()
{...9 lines }

private void define_follows(Node pivot)
{...32 lines }

private void transition_table()
{...7 lines }

private void get_transition(Terminal pivot)
{...30 lines }

private boolean verificate_elements(ArrayList<Integer> elements)
{...7 lines }

private int get_index_terminal(ArrayList<Integer> elements)
{...7 lines }

private String get_firts(Node node)
{...8 lines }

private String get_lasts(Node node)
{...8 lines }

public void tree_graph()
{...25 lines }

private void tree_graph(Node node)
{...53 lines }
```

```
public void graph_table_follows()
{...16 lines }

public void graph_table_transitions()
{...35 lines }

public void graph_automaton()
{...15 lines }

private String get_String_array(ArrayList<Integer> elements)
{...10 lines }

public static void save_file(String cadena,String name)
{...18 lines }

public void analize_lexeme(String lexeme)
{...25 lines }

private boolean verificate_char(char actual, String nonterminal)
{...3 lines }

private String verificate_transitions(String origin, char character)
{...14 lines }

private boolean aceptation(String origin)
{...13 lines }
```

Set

Este objeto contiene los siguientes atributos

- String pattern: Este atributo guarda el patrón que tiene el conjunto.
- String lexical_component: Guarda el nombre del patrón.
- ArrayList<Character> elements1: Guarda un arreglo de caracteres individuales que se definen en la entrada.
- Interval elements2;. Almacena un origen y un destino.

Los métodos con los que cuentan los objetos son :

- check_element: Este método recibe el carácter el cual será buscado en el arreglo de caracteres que pertenecen a dicho conjunto.
- Analize_Pattern: Simplemente crea los intervalos o crea la lista de elemento en base al patron. definido para dicho conjunto.