

Universidad de San Carlos de Guatemala
Facultad de ingeniería
Escuela de Ciencias y Sistemas
Organización de Lenguajes y Compiladores 1
Segundo semestre 2022
Ing. Kevin Adiel Lajpop Ajpacaja
Tutor Académico Moises Gonzalez Fuentes

Manual Técnico

Jhonatan Josué Tzunun Yax

Introducción

En el siguiente manual se describe a detalle las clases. Así como lo que contiene cada una y porque fue necesaria la creación de esta, para que se usó cada método, variable y lista, explicando de forma breve la lógica aplicada en este proyecto.

Listado de clases y archivos

Paquete **AST**:

- Nodo
- Python
- TablaError

Paquete **Analizadores**:

- Generador
- lex
- parser

Paquete **Errores**:

- Error_
- ListaError

Paquete **Instrucciones**:

- Instruction

Paquete **AST**:

Nodo:

- **Nodo**: Constructor de la clase nodo.
- **getNombre**: Obtiene el valor de la variable nombre.
- **setNombre**: Ingresa un valor a la variable nombre.
- **getHijos**: Obtiene los hijos del nodo.
- **getValor**: Obtiene el valor del nodo.
- **setValor**: Ingresa un valor a la variable valor.
- **getIdNodo**: Obtiene el id del nodo.
- **setIdNodo**: Ingresa un valor a la variable id.
- **addHijo**: Ingresa un hijo a la lista de hijos.

Python:

- **Python**: Constructor de la clase nodo.
- **replaceLast**: Busca y reemplaza el ultimo carácter que coincide con el valor pasado por parámetro.
- **Tab**: Recibe un entero como parámetro y retorna un String con el numero de tabulaciones ingresadas.
- **EXPRESSION**: Devuelve un String con la sintaxis de python de una expresión.
- **OP_RELACIONAL**: Devuelve un String con la sintaxis de Python de una operación relacional.
- **REL**: Devuelve un String con la sintaxis de Python de una operación relacional o expresión.
- **LISTREL**: Devuelve un String con una lista de REL.
- **VAR**: Devuelve un String con la sintaxis de Python de una variable o variables.
- **STATEMENT**: Devuelve un String con la sintaxis de Python de una declaración de variable.
- **ASSIGNMENT**: Devuelve un String con la sintaxis de Python de una asignación de variable.
- **CONDITION**: Devuelve un String con la sintaxis de Python para una condición.
- **CONDITIONALS**: Devuelve un String con la sintaxis de Python para una condición multiple:
- **CONDITIONAL**: Devuelve un String con la sintaxis de Python para la declaración de un condicional.
- **DEFAULT**: Devuelve un String con la sintaxis de Python para un else.
- **C**: Devuelve un String con la sintaxis de Python para un caso de un switch.
- **CASE**: Llama a la función C y devuelve un String.
- **SWITCH**: Devuelve un String con la sintaxis de Python para un switch.
- **INCREASE**: Devuelve un String con la sintaxis de Python para declarar un incremento en un For de Python.
- **INSTRUCCIONES**: Llama a la función INSTRUCTIONS o INSTRUCTION, esta es opcional.

- FOR: Devuelve un String con la sintaxis de Python para un for.
- WHILE: Devuelve un String con la sintaxis de Python para un while.
- REPETIR: Devuelve un String con la sintaxis de Python para un do while.
- RETURN: Devuelve un String con la sintaxis de Python para retornar una variable o expresion.
- PARAMETRO: Es llamada por la función LISTAPARAMETROS y retorna un String.
- LISTAPARAMETROS: Devuelve un String con la sintaxis de Python de una lista de parámetros.
- PARAMS: Devuelve un String con la lista de parámetros.
- METODO: Devuelve un String con la sintaxis de Python para un método.
- FUNCION: Devuelve un String con la sintaxis de Python para una función.
- PARAMETROS: Devuelve un String con la sintaxis de Python para un lista de parámetros al ejecutar una función.
- EXEC: Devuelve un String con la sintaxis de Python para ejecutar una función.
- PRINT: Devuelve un String con la sintaxis de Python para imprimir el retorno de una función o una expresion.
- INSTRUCTION: Busca coincidencias en los hijos de la raíz del árbol y llama a las demás funciones.
- INSTRUCTIONS: Llama a la función INSTRUCTION.
- GLOBAL: Llama a la función INSTRUCTIONS y crea el entorno de Python.
- createPyFile: Crea el archivo .py.

TablaError:

- createHtmlTable: Crea la estructura del archivo html para una tabla.
- createHtmlFile: Crea un archivo .html.

Paquete Analizadores:

- Generador: genera los archivos Parser.java y Scanner.java y Símbolos.java.
- Lex: Se crea el análisis léxico
- Parser: Se crea el análisis sintáctico.

Paquete Errores:

Error_:

- Error_: Constructor de la clase Error.
- getMessage: retorna el mensaje.
- setMessage: setea el mensaje.
- getType: retorna el tipo.

- setType: setea el tipo.
- getLine: retorna la línea.
- setLine: setea la línea.
- getColumn: retorna la columna.
- setColumn: setea la columna.

ListaError:

- addError: añade un error a la lista.

Paquete Instrucciones:

Instructions:

- analyze: Lee la entrada y lo pasa por el analizador léxico y sintactico.
- getInstance: Devuelve la instancia si está creada y si no la crea.
- getListError: Devuelve la lista de error.
- setListError: Setea la lista de error.
- getGraphNode: Crea los nodos para graphviz.
- getDot: Crea la estructura de dot.
- createASTGraph: Crea el archivo .dot y crea la imagen.

Link github: <https://github.com/TheJhonXD/OLC1-201900831.git>