Manual Tecnico

Para programar la aplicacion de JPR se hizo uso de la programacion orientada a objetos y de las clases abstractas para facilitar las operaciones necesarias del interprete. Para esto se hizo uso de las clases abstractas la cual se implemento en el archive instruccion.py, ya que como todas las instrucciones hiban a tener un metodo interpreter y otro de getNodo(para el AST) estas heredan de esta clase abstracta.

En el apartado de instrucciones se crearon los siguientes archivos:

- Asignacion
- Break
- Case
- Continue
- DeclaracionArr1
- DeclaracionArr2
- Decremento
- Incremento
- For
- Funcion
- If
- Imprimir
- Incremento
- Llamada
- Main
- ModificarArr
- Return
- ModificarArr
- Return
- Switch
- While

Como las expresiones también heredan de la clase abstracta y las clases que heredan de esta son:

- AccesoArr
- Aritmética
- Casteo
- Identificador
- Lógica
- Primitivos
- Read
- Relacional

Otras clases que heredan de instrucción son las funciones nativas, estas funciones se crean al ejecutarse el programa ya que so funciones que forman parte de JPR.

- Length
- Round
- ToLower
- ToUpper
- Truncate
- TypeOf

Para manejar la información de las variables se usaron algunas clases como:

Tabla Símbolos:

> Esta clase almacena en un diccionario las variables que se declaren y también es como una lista ya que tiene un apuntador a tabla anterior

Tipo :

> En este archivo se crean 4 clases que es tipo, operador aritmético, relacional y lógico con el fin de poder llevar un control de tipo de variable u operador se este usando.

Símbolo:

> En esta clase se guardara la información de una variable.

Exception:

> Esta clase ayudara a capturar los error ya que al llamarlo se le pasa como parámetro un tipo de error, una descripción, la fila y la columna.

Árbol:

> Aquí se guardaran las instrucciones, una lista de funciones donde estarán las nativas por defecto y las que cree el usuario, una lista de excepciones de los errores capturados, la consola que es donde se mostrara la salida y el dot para generar el árbol.

Para los reportes hay una carpeta llamada Reportes en donde en archivo reportes.py se encuentran 2 funciones, la función createHTML la cual crea un archivo HTML para generar el reporte de errores. Y la otra función llamada createHTML_TS es la encargada de crear la tabla de símbolos en un archivo HTML.

Uno de los archivos mas importantes es el de gramática.py ya que aquí contiene el analizador léxico y sintactico con ayuda del uso de PLY.

Gramática.py

En la parte del analizador léxico se declararon varios tokens y palabras reservadas que servirían para poder realizar el analizador sintactico haciendo uso de estos para poder armar las producciones en base a la gramática. Acontinuacion se muestran los tokens y palabras reservadas utilizadas para la gramática de JPR.

Tokens:

'IGUAL', 'DIFERENTE', 'MENOR', 'MAYOR', 'MENIGUAL', 'MAYIGUAL', 'IGUALIGUAL',
'MAS', 'MENOS', 'MASMAS', 'MENOSMENOS', 'POR', 'DIV', 'MOD', 'POT',
'PTCOMA', 'DOSPUNTOS', 'COMA', 'COMILLASIMPLE', 'COMILLADOBLE',
'DECIMAL', 'ENTERO', 'CADENA', 'BOOL', 'CHAR', 'ID',
'PARA', 'PARC', 'CORA', 'CORC', 'LLAVEA', 'LLAVEC',
'COMENTARIO', 'COMENTARIOMULTI',
'OR', 'AND', 'NOT'

Reservadas:

'var' 'RVAR', 'int' 'RINT', 'double' 'RDOUBLE', 'boolean' 'RBOOL', 'char' 'RCHAR', 'string' 'RSTRING',

'main' 'RMAIN', 'read' 'RREAD', 'print' 'RPRINT', 'continue' 'RCONTINUE', 'return' 'RRETURN', 'new':

'switch' 'RSWITCH', 'case' 'RCASE', 'default' 'RDEFAULT', 'break' 'RBREAK',

'toLower' 'RTOLOWER', 'toUpper' 'RTOUPPER',

'while' 'RWHILE', 'for' 'RFOR',

'true' 'RTRUE', 'false' 'RFALSE'

'if' 'RIF', 'else' 'RELSE',

'null' 'RNULL',

'func' 'RFUNC',

'RNEW'

Para algunos tokens fue necesario hacer uso de las expresiones regulares para poder formarlos.

Comentario Multilinea:	r'\#*(.|\n)*?*\#'
Comentario Normal:	r'\#.*\n?'
Decimal:	r'\d+\.\d+'
Entero:	r'\d+'
Identificador:	r'[a-zA-Z][a-zA-Z_0-9]*'

Cadena: r'\"((\\\")|.|\n)*?\"'

Char: r'(\'(\\(n|t|\\|\"|\')|.)\')'

Las producciones mas importantes utilizadas en la gramática son:

Init:

Esta es como una producción auxiliar la cual contiene una lista de instrucciones.

Instrucciones:

Contiene una lista de instrucciones.

Instruccion:

En esta esta la mayoría de cosas ya que una instrucciones deriva en una declaración, asignación, incremento, decremento, sentencias de control, sentencias ciclicas, función, llamada a función, etc.

Expresión:

Aquí se agregaron todo tipo de operaciones que se pudieran realizar así como operaciones aritméticas, relacionales, lógicas, darte algún tipo a alguna variable, una llamada a función, etc.

Fin:

Esta producción ayudo bastante ya que en esta producción puede venir (;) o no entonces para validar que se pudiera terminar una instrucción con (;) o no solo es necesario agregarlo al final de la instrucción.

Para terminar de analizar el funcionamiento del interprete se explicara como se ejecuta todo, para esto se uso un método el cual se encuenta en gramática.py llamado generateCode, al llamar a este método recibe como parámetro un string el cual será la entrada de nuestro editor, se obtienen las instrucciones por medio de ply, se crea un nuevo árbol y una nueva tabla de símbolos.

Se hace una primera pasada de las instrucciones para obtener las declaraciones, funciones y asignaciones que estén fuera del main ya que esta es la única que se ejecutara, en la segunda pasada se ejecuta el main y se verifica que no hayan 2 main repetidos y por ultimo se hace la tercera pasada para ver si hay sentencias fuera del main las cuales se tomaran como errores semánticos. Y por ultimo se retorna la consola.

Debugger:

Para el debugger la lógica funciona de la misma manera que la ejecución normal con la diferencia que se ejecuta paso a paso cada instrucción, al activar el debugger lo que se hace primero es la tercera pasada que se hace en la ejecución normal, después conforme se presione el botón de Next se retira de la lista de instrucciones el primer elemento, se verifica que este elemento es una instancia del Main si lo es se guarda en una variable y sigue ejecutando lo demás verificando que sea una declaración o asignación en donde esto seria la primera pasada. Y cuando la lista de instrucciones llegue a 0 se procede a ejecutar una a una las instrucciones que tenga el Main.