Universidad LaSalle Arequipa

Facultad de Ingenieria

Escuela Profesional de Ingenieria de Software

PRACTICA 12

Curso:

Compiladores

Semestre:

VI

Integrantes:

Elmerson Ramith Portugal Carpio

Aron Josue Hurtado Cruz

Frank Jhoseph Duarte Oruro

2022

Ejercicios 5.1

Defina una clase o estructura para representar la tabla de sımbolos. Recuerde que en la tabla de simbolos usted debe almacenar:

- Identificadores
- Funciones o metodos
- Objetos

Normalmente un dato de la tabla de sclaro ımbolos deberıa contener:

- Token y lexema
- Posicion del token
- En que funcion fue creado
- Informacion adicional que usted considere importante

Ademas, esta tabla de simbolos deberıa tener metodos/funciones para buscar, insertar e eliminar un elemento.

```
class analizador:
  def __init__(self, lexema, tipo, categoria , funcion_padre, line = None):
    self.lexema = lexema
    self.tipo = tipo
    self.categoria = categoria
    self.funcion_padre = funcion_padre
    self.line = line

array = [ ]
array_muerte = [ ]
funcion = "def"

def encontrar(lexema):
  valor = False
  for symbol in array:
    if symbol.lexema == lexema:
      valor = True
  return valor

def agregar(lexema, tipo, categoria, funcion_padre):
    node_symbol = analizador(lexema, tipo, categoria, funcion_padre)
    array.append(node_symbol)
```

Ejercicios 5.2

Defina una funcion que permita recorrer un arbol sint actico desde la raiz a sus hijos. Mientras que se recorre el arbol, debera buscar/insertar/elimnar cada identificador encontrado en la tabla de simbolos segun corresponda e verificar el scope de estos identificadores. Si detecta un error referente al scope, debera mostrar mensajes de error detallando el tipo de error.

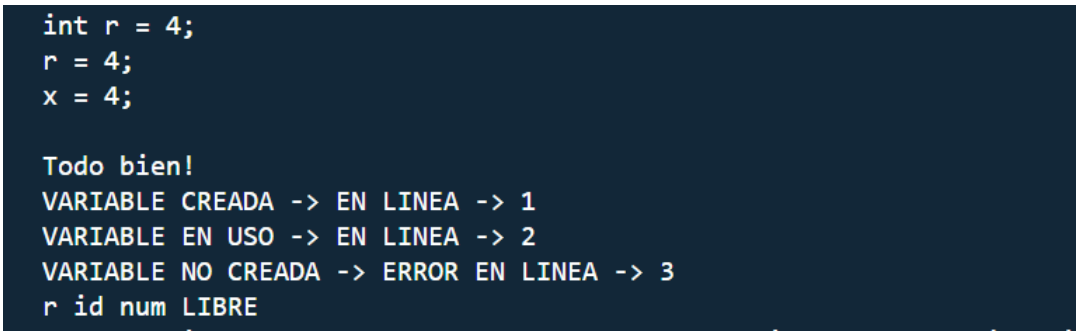
```
# Creacion de variables-----------------------------------------------------
  if (root.symbol.symbol == 'DECLARATION'):
      variable = root.children[1]
      nodo_tipo = root.children[0]
      expresion = identificado(root)
      aux = root

      # Vamos a comprovar si esta o no es una funcion-----------------
      while aux.symbol.symbol != 'FUNCTION':
        if aux.father == None:
          break
        aux = aux.father

      padre_asigando = "LIBRE"
      ## tomanos el nombre de la funcion perteneciente
      if aux.symbol.symbol == 'FUNCTION':
        padre_asigando = aux.children[1].lexeme

      if encontrar(variable.lexeme):
        print("VARIABLE YA CREADA -> ERROR EN LINEA ->", variable.line)
      else:
        print("VARIABLE CREADA -> EN LINEA ->", variable.line)
        if nodo_tipo.children[0].lexeme == "bool" and  expresion == "BOOLEAN" :
            tipo = "id"
            categoria = expresion
            padre = padre_asigando
            agregar(variable.lexeme, tipo, categoria, padre)
```

```
int r = 4;
r = 4;
x = 4;
Todo bien!
VARIABLE CREADA -> EN LINEA -> 1
VARIABLE EN USO -> EN LINEA -> 2
VARIABLE NO CREADA -> ERROR EN LINEA -> 3
r id num LIBRE
```



Repositorio del trabajo

Aqui esta el repositorio Link