

Descripción de los nodos del Árbol Abstracto (AST) mediante una Gramática Abstracta.

CATEGORIES

expresion, sentencia, tipo, definicion

NODES

programa -> definicion*;

defVariable:definicion -> ident:string tipo;

defStruct:definicion -> ident:string cuerpoStruct*;

cuerpoStruct -> ident:string tipo;

defFuncion:definicion -> ident:string parametrosFuncion:defVariable* tipo

definiciones:defVariable* sentencias:sentencia*;

tipoEntero:tipo -> ;

tipoReal:tipo -> ;

tipoChar:tipo -> ;

tipoArray:tipo -> dimension:string tipo;

tipoStruct:tipo -> ident:string;

tipoVoid:tipo -> ;

return:sentencia -> expresion;

asignacion:sentencia -> izq:expresion der:expresion;

print:sentencia -> expresion;

printsp:sentencia -> expresion;

println:sentencia -> expresion;

read:sentencia -> expresion;

bucleWhile:sentencia -> condicion:expresion cuerpo:sentencia*;

sentenciaCondicional:sentencia -> condicion:expresion cuerpoIf:sentencia*

cuerpoElse:sentencia*;

invocacionFuncionSentencia:sentencia -> id:string parametros:expresion*;

accesoStruct:expresion -> expresion ident:string;

accesoArray:expresion -> ident:expresion posicion:expresion;

variable:expresion -> ident:string;

literalInt:expresion -> value:string;

literalReal:expresion -> value:string;

literalChar:expresion -> value:string;

cast:expresion -> tipo expresion;

negacion:expresion -> expresion;

expresionAritmetica:expresion -> izquierda:expresion operador:string

derecha:expresion;

expresionBinaria:expresion -> izquierda:expresion operador:string

derecha:expresion;

expresionLogica:expresion -> izquierda:expresion operador:string

derecha:expresion;

invocacionFuncionExpresion:expresion -> id:string parametros:expresion*;