

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

CATEGORIES

declaraciones;

tipo;

sentencia;

expression;

NODES

program -> declaraciones*;

declaracionstructs:declaraciones -> nombre:string
definicion*;

declaracionglobales:declaraciones -> definicion;

declaracionfuncion:declaraciones -> nombre:string
argumento:definicion* tipo? variablesLocales:definicion*
sentencia*;

definicion -> IDENT:string tipo;

printSentencia:sentencia -> expression*;

readSentencia:sentencia -> expression*;

printspSentencia:sentencia -> expression*;

printlnSentencia: sentencia -> expression*;

funcionSentencia:sentencia -> nombre:string

argumento:expression*;

asignacionSentencia:sentencia -> left:expression
expression;

returnSentencia:sentencia -> expression? ;

ifSentencia:sentencia -> condicion:expression

entonces:sentencia* otro:sentencia*;

whileSentencia:sentencia -> condicion:expression
entonces:sentencia*;

intExpresion:expression -> int;
realExpresion:expression -> double;
identificadorExpresion: expression -> string;
charExpresion:expression -> string;
accesoArrayExpresion:expression -> acceso:expression
indice:expression;
parentesisExpresion:expression -> expression;
castExpresion:expression -> tipo expression;
negacionExpresion:expression -> expression;
arithmeticExpresion:expression -> left:expression
operator:string right:expression;
logicExpresion:expression -> left:expression
operator:string right:expression;
boolExpresion:expression -> left:expression
operator:string right:expression;
acederCap:expression -> left:expression right:string;
funcionExpresion:expression -> nombre:string
argumentos:expression*;

intTipo:tipo -> string;
floatTipo:tipo -> string;
charTipo:tipo -> string;
arrayTipo:tipo -> tamArray:int tipo;
StringTipo:tipo-> string;
voidTipo:tipo -> ;

ATTRIBUTE GRAMMAR Identification

funcionSentencia -> declaracionfuncion;
funcionExpresion -> declaracionfuncion;
identificadorExpresion -> definicion;
StringTipo -> definicion*;

sentencia -> [inherited] declafuncion:declaracionfuncion;

ATTRIBUTE GRAMMAR TypeChecking

expression -> tipoexpresion:tipo;

expression -> lvalue:boolean;

declaracionfuncion -> tipofunc:tipo;

funcionSentencia -> tipofunc:tipo;

ATTRIBUTE GRAMMAR MemoryAllocation

definicion -> [inherited] address:int;

CODE SPECIFICATION Map1

run[program]

define[declaraciones]

ejecuta[sentencia]

valor[expression]

direccion[expression]