

Gramática Abstracta

CATEGORIES

tipo, definicion, sentencia, expresion

NODES

programa -> definiciones:definicion*;

tipoInt:tipo -> ;

tipoFloat:tipo -> ;

tipoChar:tipo -> ;

tipoVoid:tipo -> ;

tipoArray:tipo -> tipo dimensiones:string;

tipoStruct:tipo -> nombre:string;

defVariable:definicion -> tipo nombre:string ambito:string;

defAtributo:definicion -> tipo nombre:string;

defStruct:definicion -> nombre:string atributos:defAtributo*;

defFuncion:definicion -> nombre:string parametros:parametro*
retorno:tipo declaraciones:defVariable* sentencias:sentencia*;

parametro -> tipo nombre:string;

llamadaFuncionExpresion:expresion -> nombre:string args:expresion*;

asignacion:sentencia -> variable:expresion valor:expresion;

read:sentencia -> variable:expresion;

if:sentencia -> condicion:expresion cierto:sentencia*
falso:sentencia*;

while:sentencia -> condicion:expresion sentencias:sentencia*;

print:sentencia -> expresion;

printsp:sentencia -> expresion;

println:sentencia -> expresion;

llamadaFuncionSentencia:sentencia -> nombre:string args:expresion*;
return:sentencia -> expresion;
returnVacio:sentencia -> ;

expresionAritmetica:expresion -> izquierda:expresion operador:string
derecha:expresion;

expresionBooleana:expresion -> izquierda:expresion operador:string
derecha:expresion;

expresionComparativa:expresion -> izquierda:expresion operador:string
derecha:expresion;

negacionBooleana:expresion -> expresion;

accesoArray:expresion -> izquierda:expresion derecha:expresion;

accesoStruct:expresion -> expresion campo:string;

variable:expresion -> name:string;

cast:expresion -> tipo expresion;

constanteInt:expresion -> valor:string;

constanteFloat:expresion -> valor:string;

constanteChar:expresion -> valor:string;