## Attribute Grammar

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| programa → definiciones:definicion\* |  |  |
| variable → nombre:String tipado:tipo | Asignable=true |  |
| variablesStruct → variables:variable | Asignable=true |  |
| variablesParametros → variables:variable\* |  |  |
| returnTipo → tipo:tipo |  |  |
| variablesDefinidas → variablesdefinidas:variablesDefinidas\* |  |  |
| arrayDefinido → nombre:String type:tipo |  |  |
| parametrosPasados → params:expresion\* |  |  |
| sentencias → list:sentencia\* |  |  |
| sentenciaCondicional → condicional:sentencia |  |  |
|  |  |  |
| variableDefinida:definicion → var:variable |  |  |
| estructuraDefinida:definicion → nombre:String variables:variable\* |  |  |
| metodoDefinido:definicion → parametros:variable\* retorno:tipo variablesDef:variableDefinida\* sentenciasEj:sentencia\* | Tipo=retorno |  |
|  |  |  |
| expresionParentesis:expresion → expr:expresion | Tipo=exp.tipo |  |
| expresionIdent:expresion → identificador:String | Tipo=identificador.tipo |  |
| expresionConstante:expresion → constante:String | Tipo=constante.tipo |  |
| expresionLlamadaMetodo:expresion → nombre:String parametros:expresion\* | Tipo=nombre.tipo | comprobarParams(nombre,parámetros) |
| expresionCampoStruct:expresion → nombreStruct:expresion campo:String | Tipo=campo.tipo |  |
| expresionArray:expresion → id:expresion pos:expresion | Tipo=id.tipo | Pos.tipo==Int |
| expresionAritmetica:expresion → l:expresion op:String r:expresion | Tipo=l.tipo | l.tipo=r.tipo |
| expresionLogica:expresion → l:expresion op:String r:expresion | Tipo=boolean |  |
| expresionCast:expresion → type:tipo ex:expresion | Tipo=type |  |
|  |  |  |
| sentenciaExpresion:sentencia → expr:expresion |  |  |
| sentenciaAsignacion:sentencia → l:expresion r:expresion |  | l.tipo==r.tipo  l.asignable==true | l.tipo==r.tipo  l.asignable==true |
| sentenciaIf:sentencia → expr:expresion ifTrue:sentencia\* |  | Expr.tipo=bool |
| sentenciaIfElse:sentencia → expr:expresion ifTrue:sentencia\* ifFalse:sentencia\* |  | Expr.tipo=bool |
| sentenciaWhile:sentencia → expr:expresion contenido:sentencia\* |  | Expr.tipo=bool |
| sentenciaPrint:sentencia → expr:expresion |  |  |
| sentenciaRead:sentencia → expr:expresion |  |  |
| sentenciaReturn:sentencia → expr:expresion |  | Expr.tipo==método.tipo |
| sentenciaPrintVoid:sentencia → λ |  |  |
| sentenciaReturnVoid:sentencia → λ |  |  |
|  |  |  |
| tipoInt:tipo → λ |  |  |
| tipoFloat:tipo → λ |  |  |
| tipoChar:tipo → λ |  |  |
| tipoStruct:tipo → nombre:String |  |  |
| tipoArray:tipo → dim:int type:tipo | Tipo=type |  |
|  |  |  |

Recordatorio de los operadores (para cortar y pegar): ⇒ ⇔ ≠ ∅ ∈ ∉ ∪ ∩ ⊂ ⊄ ∑ ∃ ∀

### Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nodo/Categoría Sintáctica** | **Nombre del Atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
| expresion | tipo | Tipo | Heredado | Tipo devuelto por la expresión |
| definicion | tipo | Tipo | Sintetizado | Tipo devuelto por el método definido |
| variable | asignable | Bool |  |  |

Bool comprobarParams(ident, parámetros\*){

For(i<parámetros.size){

If(ident.params[i].tipo == parametros[i].tipo)

Continue;

Else

Return false;

}

Return true;

}