**Attribute Grammar**

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| **programa** → *definicion*:definicion\* |  |  |
| **cuerpoStruct** → *ident*:String *tipo*:tipo | cuerpoStruct[ident]==null | cuerpoStruct[ident]=defCampo |
|  |  |  |
| **defVariable**:definicion → *ident*:String *tipo*:tipo | variables.nombre==null | variables[nombre]=defVariable |
| **defStruct**:definicion → *ident*:String *cuerpostruct*:cuerpoStruct\* | estructuras[ident]==null | estructuras[ident]=defStruct  {  visit(cuerpoStruct i)  } |
| **defFuncion**:definicion → *ident*:String *parametrosFuncion*:defVariable\* *tipo*:tipo *definiciones*:defVariable\* *sentencias*:sentencia\* | funciones[ident]==null | funciones[ident]=defFuncion  {  variables.set()  visit(parametrosFuncion i)  visit(cuerpo, funciones[ident])  variables.reset()  } |
|  |  |  |
| **tipoEntero**:tipo → λ |  |  |
| **tipoReal**:tipo → λ |  |  |
| **tipoChar**:tipo → λ |  |  |
| **tipoArray**:tipo → *dimension*:String *tipo*:tipo |  |  |
| **tipoStruct**:tipo → *ident*:String | estructuras[ident] ≠ null | tipoStruct.definicion=estructuras[ident] |
| **tipoVoid**:tipo → λ |  |  |
|  |  |  |
| **return**:sentencia → *expresion*:expresion |  | return.funcionEnLaQueEstoy = DefFuncion |
| **asignacion**:sentencia → *izq*:expresion *der*:expresion |  |  |
| **print**:sentencia → *expresion*:expresion |  |  |
| **read**:sentencia → *expresion*:expresion |  |  |
| **bucleWhile**:sentencia → *condicion*:expresion *cuerpo*:sentencia\* |  | bucleWhile.funcionEnLaQueEstoy = DefFuncion |
| **sentenciaCondicional**:sentencia → *condicion*:expresion *cuerpoIf*:sentencia\* *cuerpoElse*:sentencia\* |  | sentenciaCondicional.funcionEnLaQueEstoy = DefFuncion |
| **invocacionFuncionSentencia**:sentencia→ *id*:String *parametros*:expresion\* | funciones[id] ≠ null | invocacionFuncionSentencia.definicion = funciones[nombre]  invocacionFuncionSentencia.funcionEnLaQueEstoy = DefFuncion |
|  |  |  |
| **accesoStruct**:expresion → *expresion*:expresion *ident*:String |  |  |
| **accesoArray**:expresion → *ident*:expresion *posicion*:expresion |  |  |
| **variable**:expresion → *ident*:String | variables.nombre≠null | variable.definicion=variables(nombre) |
| **literalInt**:expresion → *value*:String |  |  |
| **literalReal**:expresion → *value*:String |  |  |
| **literalChar**:expresion → *value*:String |  |  |
| **cast**:expresion → *tipo*:tipo *expresion*:expresion |  |  |
| **negacion**:expresion → *expresion*:expresion |  |  |
| **expresionAritmetica**:expresion → *izquierda*:expresion *operador*:String *derecha*:expresion |  |  |
| **expresionBinaria**:expresion → *izquierda*:expresion *operador*:String *derecha*:expresion |  |  |
| **invocacionFuncionExpresion:**expresion→*id***:**String*parametros***:**expresion\* | funciones[id] ≠ null | invocacionFuncionExpresion.definicion = funciones[nombre] |
|  |  |  |

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

**Atributos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoría Sintáctica** | **Nombre del atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
| invocacionFuncion | definicion | DefFuncion | Sintetizado |  |
| variable | definicion | DefVariable | Sintetizado |  |
| tipoStruct | definicion | DefStruct | Sintetizado |  |