Attribute Grammar

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| programa → definiciones:definicion\* |  |  |
|  |  |  |
| definicionFuncion:definicion → nombre:String params:definicionVariable\* tipo:tipo variablesLocales:definicionVariable\* sentencias:sentencia\* |  |  |
| definicionVariable:definicion → nombre:String tipo:tipo |  |  |
| definicionStruct:definicion → nombre:String campos:definicionCampo\* |  |  |
| definicionCampo:definicion → nombre:String tipo:tipo |  |  |
|  |  |  |
| tipoEntero:tipo → λ |  |  |
| tipoReal:tipo → λ |  |  |
| tipoChar:tipo → λ |  |  |
| tipoArray:tipo → longitud:int tipo:tipo |  |  |
| tipoStruct:tipo → nombre:String |  |  |
| tipoVoid:tipo → λ |  |  |
|  |  |  |
| print:sentencia → expresion:expresion tipo\_print:String | esTipoPrimitivo(expresion.tipo) |  |
| read:sentencia → expresion:expresion | esTipoPrimitivo(expresion.tipo) |  |
| asignacion:sentencia → izquierda:expresion derecha:expresion | esTipoPrimitivo(derecha.tipo)  mismoTipo(izquierda.tipo, derecha.tipo) |  |
| if:sentencia → condicion:expresion verdadero:sentencia\* falso:sentencia\* |  |  |
| while:sentencia → condicion:expresion sentencia:sentencia\* |  |  |
| invocacion:sentencia → nombre:String params:expresion\* | coincidenTiposParametros(nombre, params) |  |
| return:sentencia → expresion:expresion\* | return.definicionFuncion.tipo = expresion.tipo |  |
|  |  |  |
| constanteEntero:expresion → valor:int |  | constanteEntero.tipo = tipoEntero  constanteEntero.modificable = false |
| constanteReal:expresion → valor:double |  | constanteReal.tipo = tipoReal  constanteReal .modificable = false |
| constanteChar:expresion → valor:String |  | constanteChar.tipo = tipoChar  constanteChar modificable = false |
| variable:expresion → nombre:String |  | variable.tipo = variable.definicion.tipo  variable.modificable = true |
| expresionAritmetica:expresion → izq:expresion operador:String der:expresion | esTipoPrimitivo(izq.tipo)  esTipoPrimitivo(der.tipo)  izq.tipo ≠ tipoChar  der.tipo ≠ tipoChar | expresionAritmetica.tipo = izq.tipo  expresionAritmetica.modificable = false |
| expresionLogica:expresion → izq:expresion operador:String der:expresion | izq.tipo = tipoEntero  der.tipo = tipoEntero | expresionLogica.tipo = izq.tipo  expresionLogica.modificable = false |
| comparacion:expresion → izq:expresion operador:String der:expresion | esTipoPrimitivo(izq.tipo)  esTipoPrimitivo(der.tipo)  izq.tipo ≠ tipoChar  der.tipo ≠ tipoChar |  |
| expresionUnaria:expresion → expresion:expresion operador:String | expresion.tipo = tipoEntero | expresionUnaria.tipo = expresion.tipo  expresionUnaria.modificable = false |
| conversion:expresion → nuevoTipo:tipo expresion:expresion | esTipoPrimitivo(nuevoTipo)  esTipoPrimitivo (expresion.tipo)  ~ mismoTipo(nuevoTipo, expresion.tipo) | conversion.tipo = nuevoTipo  conversion.modificable = false |
| invocacionExpresion:expresion → nombre:String params:expresion\* | coincidenTiposParametros(params) | invocacionExpresion.tipo = nombre.tipo  invocacionExpresion.modificable = false |
| accesoArray:expresion → array:expresion indice:expresion | indice.tipo = tipoEntero  array.tipo = tipoArray | accesoArray.tipo = array.tipo  accesoArray.modificable = true |
| accesoCampo:expresion → struct:expresion campo:expresion |  | accesoCampo.tipo = struct.tipo  accesoCampo.modificable = true |
|  |  |  |

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

Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nodo/Categoría Sintáctica** | **Nombre del Atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
| expresion | tipo | Tipo | Sintetizado | Tipo de la expresión |
| expresion | modificable | boolean | Sintetizado | Indica si la expresión puede aparecer en el lado izquierdo de una asignación |
|  |  |  |  |  |