## Attribute Grammar

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| program → bloque:bloque\* |  |  |
| atributo → name:String type:type |  |  |
| parametro → nombre:String type:type | type ∈ tiposSimples |  |
| array → pos:expresion | mismoTipo(pos.tipo, intType) |  |
| llamada → nombre:String parametros:expresion\* | |parámetros|==|llamada.definition.parametros|  mismoTipo(parametros**i**.tipo, definition.parametros**i.**tipo) |  |
|  |  |  |
| varDef:bloque → name:String type:type |  |  |
| estructuraDef:bloque → nombre:String atributos:atributo\* |  |  |
| funcionDef:bloque → name:String parametros:parametro\* type:type variables:varDef\* sentencias:sentencia\* | type ∈ tiposSimples | sentencia**i**.inFunc = funcionDef |
|  |  |  |
| intType:type → λ |  |  |
| realType:type → λ |  |  |
| charType:type → λ |  |  |
| arrayType:type → size:expresion type:type | mismoTipo(size.tipo, intType) |  |
| customType:type → nombre:String |  |  |
|  |  |  |
| read:sentencia → value:expresion | value.tipo ∈ tiposSimples  value.modificable |  |
| print:sentencia → operacion:string value:expresion | value.tipo != null  value.tipo ∈ tiposSimples |  |
| assign:sentencia → left:expresion right:expresion | left.tipo ∈ tiposSimples  mismoTipo(left.type, right.type)  left.modificable |  |
| whileloop:sentencia → condicion:expresion sentencias:sentencia\* | mismoTipo(condición.tipo, intType) | sentencias**i**.inFunc = whileloop.inFunc |
| ifDef:sentencia → condicion:expresion iftrue:sentencia\* iffalse:sentencia\* | mismoTipo(condición.tipo, intType) | iftrue**i**.inFunc = ifDef.inFunc  iffalse**i**.inFunc = ifDef.inFunc |
| returnSent:sentencia → retorno:expresion | ***Si returnSent.inFunc.tipo == null***  retorno==null  ***Si no***  mismoTipo(retorno.tipo, returnSent.definition.tipo) |  |
| llamadaS:sentencia → invocacion:llamada |  |  |
|  |  |  |
| variable:expresion → name:String |  | variable.tipo = variable.definition.tipo  variable.modificable = true |
| accesoStruct:expresion → left:expresion right:string | mismoTipo(left.tipo, customType)  right ∈ left.tipo.definition.atributos.name | accesoStruct.tipo = left.tipo  accesoStruct.modificable = true |
| integral:expresion → valor:String |  | Integral.tipo = intType  Integral.modificable = false |
| real:expresion → valor:String |  | real.tipo = realType  real.modifcable = false |
| caracter:expresion → valor:String |  | caracter.tipo = charType  caracter.modificable = false |
| accesoArray:expresion → var:expresion dimensiones:array | mismoTipo(var.tipo, arrayType) | accesoArray.tipo = var.tipo.tipo  accesoArray.modificable = true |
| parentesis:expresion → contenido:expresion |  | parentesis.tipo = contenido.tipo  parentesis.modificable = contenido.modificable |
| casting:expresion → tipo:type contenido:expresion | tipo ∈ tiposSimples  Contenido.tipo ∈ tiposSimples  Contenido.tipo!=tipo | casting.tipo = tipo  casting.modificable = contenido.modificable |
| aritmetica:expresion → left:expresion operando:String right:expresion | Left.tipo ∈ tiposSimples  mismoTipo(left.tipo, right.tipo) | aritmetica.tipo = left.tipo aritmetica.modificable = false |
| comparativa:expresion → left:expresion operando:String right:expression | left.tipo ∈ tiposSimples  mismoTipo(left.tipo, right.tipo) | comparativa.tipo = intType  comparativa.modifcable = false |
| logica:expresion → left:expresion operando:String right:expression | mismoTipo(left.tipo, right.tipo)  mismoTipo(left.tipo, intType) | logica.tipo = intType  logica.modificable = false |
| llamadaE:expresion → invocacion:llamada | Invocacion.definition.tipo !=null | llamadaE.tipo = invocacion.definition.tipo  llamadaE.modificable = false |
|  |  |  |

Tabla de Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nodo/Categoría** | **Nombre** | **Tipo Java** | **H/S** | **Descripción** |
| expresion | tipo | Type | Sintetizado | Tipo de la expresión (operaciones que admite) |
| expresion | modificable | boolean | Sintetizado | Indica si la expresión puede aparecer a la izquierda de una asignación |
| sentencia | inFunc | FuncionDef | Heredado | Definicion de la función que contiene la sentencia |

Funciones auxiliares:

***mismoTipo****(tipoA, tipoB)* { tipoA == tipoB }

