

Attribute Grammar Fase de Identificación

Nodo	Predicados	Reglas Semánticas
Programa → <i>declaraciones</i> :Declaracion*		
DefVariable :Declaracion → <i>nombre</i> :String <i>tipo</i> :Tipo	variables[nombre] == ∅	variables[nombre] = DefVariable
DefStruct :Declaracion → <i>nombre</i> :String <i>listaCampos</i> :DefCampo*	structs[nombre] == ∅	structs[nombre] = DefStruct
DefFunction :Declaracion → <i>nombre</i> :String <i>listaParametros</i> :DefVariable* <i>tipo</i> :Tipo <i>listaDeclaraciones</i> :DefVariable* <i>listaSentencias</i> :Sentencia*	funciones[nombre] == ∅	funciones[nombre] = DefFunction
SenInvocacion :Sentencia, Expresion → <i>nombre</i> :String <i>listaArgumentos</i> :Expresion*	∃ funciones[nombre]	invocacion = funciones[nombre]
SenAsignacion :Sentencia → <i>left</i> :Expresion <i>right</i> :Expresion		
SenPrint :Sentencia → <i>expresion</i> :Expresion		
SenPrintSp :Sentencia → <i>expresion</i> :Expresion		
SenPrintLn :Sentencia → <i>expresion</i> :Expresion		
SenRead :Sentencia → <i>expresion</i> :Expresion		
SenIf :Sentencia → <i>condicion</i> :Expresion <i>sentenciasIf</i> :Sentencia* <i>sentenciasElse</i> :Sentencia*		
SenWhile :Sentencia → <i>condicion</i> :Expresion <i>sentencias</i> :Sentencia*		
SenReturn :Sentencia → <i>retorno</i> :Expresion		
ExAritmetica :Expresion → <i>left</i> :Expresion <i>operador</i> :String <i>right</i> :Expresion		
ExLogica :Expresion → <i>left</i> :Expresion <i>operador</i> :String <i>right</i> :Expresion		
ExRelacional :Expresion → <i>left</i> :Expresion <i>operador</i> :String <i>right</i> :Expresion		
ExNot :Expresion → <i>expresion</i> :Expresion		
ExInvocacion :Expresion → <i>nombre</i> :String <i>listaArgumentos</i> :Expresion*	∃ funciones[nombre]	invocacion = funciones[nombre]
ExIndice :Expresion → <i>left</i> :Expresion <i>indice</i> :Expresion		
ExCampo :Expresion → <i>struct</i> :Expresion <i>campo</i> :Expresion		
ExCast :Expresion → <i>to</i> :Tipo <i>from</i> :Expresion		
LitEntero :Expresion, Literal → <i>valor</i> :String		
LitReal :Expresion, Literal → <i>valor</i> :String		
LitCaracter :Expresion, Literal → <i>valor</i> :String		
Variable :Expresion → <i>nombre</i> :String	∃ variables[nombre]	declaracion = variables[nombre]
TipoEntero :Tipo → λ		
TipoReal :Tipo → λ		
TipoCaracter :Tipo → λ		
TipoStruct :Tipo → <i>nombre</i> :String	∃ structs[nombre]	definicion = structs[nombre]
TipoArray :Tipo → <i>tipo</i> :Tipo <i>tamaño</i> :String		
TipoVoid :Tipo → λ		

Recordatorio de operadores (para cortar y pegar): $\Rightarrow \Leftrightarrow \neq \emptyset \in \notin \cup \cap \subset \not\subset \sum \exists \forall$

Atributos

Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
Variable	declaracion	DefVariable	Sintetizado	Campo para asociar la variable con su definición
Sentencia, ExInvocacion	invocacion	DefFunction	Sintetizado	Campo para asociar una sentencia o expresión de invocación con la función que llama.
TipoStruct	definicion	DefStruct	Sintetizado	Campo para asociar la definición de un Struct con un tipo Struct

Conjuntos auxiliares

variables	Map<String, Declaracion>
funciones	Map<String, Declaracion>
structs	Map<String, Declaracion>