## Attribute Grammar

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| program → definitions:definition\* |  |  |
|  |  |  |
| varDefinition:definition → name:String type:type | variables.getFromTop(name) == null | variables[name] = varDefinition |
| structDefinition:definition → name:varType definitions:structField\* | variables.getFromAny(name) == null | variables[name] = structDefinition  structs[name] = structDefinition |
| funDefinition:definition → name:String params:definition\* return\_t:type definitions:varDefinition\* sentences:sentence\* | funciones[name] == null | funciones[name] = funDefinition |
| structField:definition → name:String type:type | variables.getFromTop(name) == null | variables[name] = structField |
|  |  |  |
| intType:type → λ |  |  |
| realType:type → λ |  |  |
| charType:type → λ |  |  |
| varType:type → type:String |  |  |
| voidType:type → λ |  |  |
| arrayType:type → size:intConstant type:type |  |  |
| structType:type → fields:structField\* |  |  |
| errorType:type → λ |  |  |
|  |  |  |
| print:sentence → expression:expression |  |  |
| printsp:sentence → expression:expression |  |  |
| println:sentence → expression:expression |  |  |
| read:sentence → expression:expression |  |  |
| assignment:sentence → left:expression right:expression |  |  |
| return:sentence → expression:expression |  |  |
| ifElse:sentence → expression:expression if\_s:sentence\* else\_s:sentence\* |  |  |
| while:sentence → expression:expression sentence:sentence\* |  |  |
| funcInvocation:sentence → name:String params:expression\* | funciones[name] ¡= null | funcInvocation.definition = funciones[name] |
|  |  |  |
| variable:expression → name:String | variables.getFromAny(name) ¡= null | variable.definition = variables[name] |
| intConstant:expression → value:String |  |  |
| realConstant:expression → value:String |  |  |
| charConstant:expression → value:String |  |  |
| voidConstant:expression → λ |  |  |
| funcInvocationExpression:expression → name:String params:expression\* | funciones[name] ¡= null | funcInvocationExpression.definition = funciones[name] |
| arithmeticExpression:expression → left:expression operator:String right:expression |  |  |
| logicalExpression:expression → left:expression operator:String right:expression |  |  |
| unaryExpression:expression → operator:String expr:expression |  |  |
| comparableExpression:expression → left:expression operator:String right:expression |  |  |
| castExpression:expression → type:type expr:expression |  |  |
| fieldAccessExpression:expression → expr:expression name:String |  |  |
| indexExpression:expression → expr:expression index:expression |  |  |
|  |  |  |

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

### Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nodo/Categoría Sintáctica** | **Nombre del Atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
| funcInvocation | definition | FuncDefinition | Sintetizado |  |
| funcInvocationExpression | definition | FuncDefinition | Sintetizado |  |
| variable | definition | VarDefinition | Sintetizado |  |