# Code Specification

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| **program** → *classnode*:classNode *methodcallsentence*:methodCallSentence |  |  |
| **classNode** → *name*:String *definition*:definition\* *createMethod*:String\* *method*:method\* |  | definitioni.address =  ∑ definitioni.type.size \* definition.name.size | 0 <= j < i |
| **method** → *name*:String *parameter*:parameter\* *retorno*:type *definition*:definition\* *sentence*:sentence\* |  | parameteri.addres = 4 + ∑ parameteri.type.size | 0 <= j < i  definitioni.address = ∑ -definitioni.type.size \* definition.name.size | 0 <= j < i |
| **parameter** → *name*:String *type*:type |  |  |
|  |  |  |
| **print**:sentence → *string*:String *expr*:expr\* |  |  |
| **read**:sentence → *expr*:expr\* |  |  |
| **assignment**:sentence → *left*:expr *right*:expr |  |  |
| **conditional**:sentence → *condition*:expr *iftrue*:sentence\* *iffalse*:sentence\* |  |  |
| **loop**:sentence → *init*:sentence\* *condition*:expr *sentence*:sentence\* |  |  |
| **returnNode**:sentence → *expr*:expr |  |  |
| **methodCallSentence**:sentence → *name*:String *args*:expr\* |  |  |
|  |  |  |
| **exprBinariaAritmetica**:expr → *left*:expr *op*:String *right*:expr |  |  |
| **exprUnariaAritmetica**:expr → *op*:String *expr*:expr |  |  |
| **exprBinariaLogica**:expr → *left*:expr *op*:String *right*:expr |  |  |
| **exprUnariaLogica**:expr → *op*:String *expr*:expr |  |  |
| **access**:expr → *left*:expr *op*:String *right*:String |  |  |
| **arrayAcces**:expr → *left*:expr *right*:expr |  |  |
| **cast**:expr → *typeToConvert*:type *expr*:expr |  |  |
| **litEnt**:expr → *string*:String |  |  |
| **litReal**:expr → *string*:String |  |  |
| **litChar**:expr → *string*:String |  |  |
| **variable**:expr → *string*:String |  |  |
| **methodCallExpr**:expr → *name*:String *args*:expr\* |  |  |
|  |  |  |
| **intType**:type → λ |  | intType.size = 2 |
| **realType**:type → λ |  | realType.size = 4 |
| **charType**:type → λ |  | charType.size = 1 |
| **arrayType**:type → *index*:int *type*:type |  | arrayType.size = arrayType.index \* arrayType.type.size |
| **voidType**:type → λ |  | voidType = 0 |
| **structType**:type → *name*:String |  | structType.size =  ∑ structType.definition.varDefinitionsi.type.size | 0 <= j < i |
|  |  |  |
| **tupleDefinition**:definition → *name*:String *vardefinition*:varDefinition\* |  | vardefinitioni.address =  ∑ vardefinitioni.type.size | 0 <= j < i |
| **varDefinition**:definition → *name*:String\* *type*:type |  |  |
|  |  |  |

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

# Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nodo/Categoría Sintáctica** | **Nombre del Atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
| Variable | definition | VarDefinition | Sintetizado | Nodo donde está definida la variable |

# Estructura de Datos Auxiliares

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo Java** | **Descripción** |

# Funciones Auxiliares