## Code Specification

|  |  |
| --- | --- |
| Functions | Code Templates |
| Run ⟦program⟧ | Run ⟦program → cuerpoPrograma\*⟧ =  #SOURCE {file}  CALL main  HALT  Prepara ⟦ cuerpoPrograma\* ⟧ |
| Prepara ⟦cuerpoPrograma⟧ | ƒ5⟦structDefinition:cuerpoPrograma → nombre:string atributos:varDefinition\* metodos:function\*⟧ =  Define⟦ atributos.declaracion ⟧  TEMPORAL  ƒ5⟦varDefinition:cuerpoPrograma → declaracion⟧ =  Define⟦ declaracion ⟧  TEMPORAL  ƒ5⟦function:cuerpoPrograma → nombre:string parametros:declaracion\* tipoRetorno:tipo cuerpo:statement\*⟧ =  Define⟦ parametros ⟧  Ejecuta⟦ cuerpo ⟧  TEMPORAL |
| Define ⟦declaracion⟧ | Define ⟦declaracion → nombre:string tipo:tipo⟧ = |
| Ejecuta ⟦statement⟧ | Ejecuta ⟦print:statement → expr\*⟧ =   #LINE {end.line}  Valor ⟦expression⟧  OUT getFormatTipo(expression.type)  Ejecuta ⟦printSp:statement → expr\*⟧ =   #LINE {end.line}  Valor ⟦expression⟧  OUT getFormatTipo(expression.type)  Ejecuta ⟦printLn:statement → expr\*⟧ =   #LINE {end.line}  Valor ⟦expression⟧  OUT getFormatTipo(expression.type)  Ejecuta ⟦asignacion:statement → left:expr right:expr⟧ =  #LINE {end.line} Direccion⟦left⟧ Valor⟦right⟧ STORE getFormatTipo (**left.type**)  Ejecuta ⟦return:statement → expr?⟧ =  Ejecuta ⟦oneExpr:statement → expr⟧ =  Ejecuta ⟦if:statement → condition:expr thenBlock:statement\* elseBlock:statement\*⟧ =  Ejecuta ⟦while:statement → condition:expr body:statement\*⟧ =  Ejecuta ⟦read:statement → expr⟧ =  Ejecuta ⟦stmtFunctionCall:statement → nombre:string expr\*⟧ = |
| Direccion ⟦expr⟧ | Direccion ⟦**variable** → nombre:string⟧ =  If global:  PUSHA {variable.declaracion.address}  Else:  PUSHA BP  PUSHA <dir>  ADD |
| Valor ⟦expr⟧ | Valor⟦fieldAccess:expr → object:expr field:string⟧ =  Valor⟦arrayAccess:expr → array:expr index:expr⟧ =  Valor⟦cast:expr → tipo:tipo target:expr⟧ =  Valor⟦not:expr → expr⟧ =  Valor ⟦expr⟧ NOT  Valor⟦operacionAritmetica:expr → left:expr operador:string right:expr⟧ =  Valor ⟦left⟧ Valor ⟦right⟧ if operador == "+"    ADD getFormatTipo( operacionAritmetica.type) if operador == "-"      SUB getFormatTipo( operacionAritmetica.type) if operador == "\*"      MUL getFormatTipo( operacionAritmetica.type) if operador == "/"       DIV getFormatTipo( operacionAritmetica.type)  Valor⟦operacionLogica:expr → left:expr operador:string right:expr⟧ =  Valor ⟦left⟧ Valor ⟦right⟧ if operador == "&&"    AND if operador == "||"      OR  Valor⟦comparacion:expr → left:expr operador:string right:expr⟧ =  Valor⟦functionCall:expr → nombre:string expr\*⟧ =  Valor⟦parentesis:expr → expr⟧ =  Valor⟦variable:expr → nombre:string⟧ =  direccion⟦variable⟧ LOAD getFormatTipo(variable.type)  Valor⟦intLiteral:expr → intValue:int⟧ =  PUSH {intValue}  Valor⟦realLiteral:expr → floatValue:float⟧ =  PUSHF {floatValue}  Valor⟦charLiteral:expr → charValue:char⟧ =  PUSHC {charValue} |