program -> defList

defList -> defListP

defListP -> def defListP

defListP -> ''

def -> varDef

def -> funDef

varDef -> var varList ;

varList -> idList

idList -> id idListCount

idListCount -> , id idListCount

idListCount -> ''

funDef -> id ( paramList ) { varDefList stmtList }

paramList -> idList

paramList -> ''

varDefList -> varDefListP

varDefListP -> varDef varDefListP

varDefListP -> ''

stmtList -> stmtListP

stmtListP -> stmt stmtListP

stmtListP -> ''

stmt -> id stmtFactorizado

stmt -> stmtIncr

stmt -> stmtDecr

stmt -> stmtIf

stmt -> stmtWhile

stmt -> stmtDoWhile

stmt -> stmtBreak

stmt -> stmtReturn

stmt -> stmtEmpty

stmtIncr -> inc id ;

stmtDecr -> dec id ;

stmtFactorizado -> = expr ;

stmtFactorizado -> ( exprList ) ;

(Creamos estas dos reglasa y borramos ‘stmt-fun-call’ porque ya no se necesita)

exprList -> expr exprListCont

exprList -> ''

exprListCont -> , expr exprListCont

exprListCont -> ''

stmtIf -> if ( expr ) { stmtList } elseIfList else

elseIfList -> elseIfListP

elseIfListP -> elif ( expr ) { stmtList } elseIfListP

elseIfListP -> ''

else -> elseTerminal { stmtList }

else -> ''

stmtWhile -> while ( expr ) { stmtList }

stmtDoWhile -> do { stmtList } while ( expr ) ;

stmtBreak -> break ;

stmtReturn -> return expr ;

stmtEmpty -> ;

expr -> exprOr

exprOr -> exprAnd exprOrP

exprOrP -> or exprAnd exprOrP

exprOrP -> ''

exprAnd -> exprComp exprAndP

exprAndP -> and exprComp exprAndP

exprAndP -> ''

exprComp -> exprRel exprCompP

exprCompP -> opComp exprRel exprCompP

exprCompP -> ''

opComp -> ==

opComp -> <>

exprRel -> exprAdd exprRelP

exprRelP-> opRel exprAdd exprRelP

exprRelP-> ''

opRel -> <

opRel -> <=

opRel -> >

opRel -> >=

exprAdd -> exprMul exprAddP

exprAddP -> opAdd exprMul exprAddP

exprAddP -> ''

opAdd -> +

opAdd -> -

exprMul -> exprUnary exprMulP

exprMulP -> opMul exprUnary exprMulP

exprMulP -> ''

opMul -> \*

opMul -> /

opMul -> %

exprUnary -> opUnary exprUnary

exprUnary -> exprPrimary

opUnary -> +

opUnary -> -

opUnary -> not

exprPrimary -> id exprPrimaryFact

exprPrimary -> array

exprPrimary -> lit

exprPrimary -> ( expr )

exprPrimaryFact -> ''

exprPrimaryFact -> ( exprList )

(Creamos estas dos reglasa y borramos ‘exprPrimary -> funCall’ porque ya no se necesita)

array -> [ exprList ]

lit -> litBool

lit -> litInt

lit -> litChar

lit -> litStr