GRAMÁTICA

Pablo Josué Barahona Luncey
202109715
3560855890101
Organización de Lenguajes y Compiladores 2

PROYECTO 1

LEXER

```
INT: 'Int';
FLOAT: 'Float';
BOOL: 'Bool';
STR: 'String';
CHAR: 'Character';

VAR: 'var';
LET: 'let';
VOID: 'void';
TRU: 'true';
FAL: 'false';
PRINT: 'print';
IF: 'if';
ELSE: 'else';
WHILE: 'while';
FOR: 'for';
IN: 'in';
SWITCH: 'switch';
CASE: 'case';
DEFAULT: 'default';
BREAK: 'break';
RETURN: 'return';
CONTINUE: 'continue';
GUARD: 'guard';
FUNC: 'func';
NIL: 'nil';
STRUCT: 'struct';
MUTATING: 'mutating';
SELF: 'self';
INOUT: 'inout';
APPEND:
'append';
REMOVELAST:
'removeLast';
REMOVE:
'remove';
AT: 'at';
ISEMPTY:
'isEmpty';
COUNT:
'count';

NUMBER : [0-
9]+ ('.'[0-9]+)?;
STRING: '"'~
["]*'"';
ID: ([a-zA-Z_])
[a-zA-Z0-9_]*;

DIF:    '!=';
IG_IG:      '==';
NOT:        '!';
OR:         '||';
AND:
'&&';
IG:       '=';
MAY_IG:   '>=';
MEN_IG:   '<=';
SUM_IG:   '+=';
SUB_IG:   '-=';
```

PROYECTO 1

LEXER

```
MAYOR: '>';
MENOR:'<';
MUL: '*';
DIV: '/';
ADD: '+';
SUB: '-';
MOD: '%';
PARIZQ: '(';
PARDER:       ')';
LLAVEIZQ:     '{';
LLAVEDER:     '}';
D_PTS:        ':';
CORIZQ:       '[';
CORDER:       ']';
COMA:         ',';
PUNTO:        '.';
COMILLA:'"';
FLECHA:'->';
GUIONBAJO:    '_';
AMP :         '&';

WHITESPACE: [ \\\r\n\t]+ ->
skip;
COMMENT : '/*' .*? '*/' -> skip;
LINE_COMMENT : '//' ~[\r\n]* ->
skip;

fragment
ESC_SEQ
  :'\\' ('\\'|'@'|'['|']'|'.'|'#'|'+'|'-'|'!'|':'| ')
  .
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

```
instruction
  : printstmt
  | ifstmt
  | declarationstmt
  | whilestmt
  | assignstmt
  | forstmt
  | guardstmt
  | breakstmt
  | continuestmt
  | fnArray
  | structCreation
  | returnstmt
  | fnstmt
  | callFunction
  ;

printstmt
  : PRINT PARIZQ expr PARDER
  | PRINT PARIZQ exprComa PARDER
  ;

ifstmt
  : IF expr LLAVEIZQ block LLAVEDER
  | IF expr LLAVEIZQ block LLAVEDER ELSE LLAVEIZQ block
LLAVEDER
  | IF expr LLAVEIZQ block LLAVEDER ELSE ifstmt
  ;
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

```
|whilestmt
  : WHILE expr LLAVEIZQ block LLAVEDER
  ;

declarationstmt
  : VAR ID D_PTS types IG expr
  | VAR ID IG expr
  | VAR ID D_PTS types
  | LET ID D_PTS types IG expr
  | LET ID D_PTS types
  | LET ID IG expr
  ;

assignstmt
  : ID IG expr
  | listAccessStruct IG expr
  | ID op=IG expr
  | ID listAccessArray IG expr
  ;

forstmt
  : FOR ID IN expr LLAVEIZQ block LLAVEDER
  | FOR ID IN exp1=expr PUNTO PUNTO PUNTO exp2=expr LLAVEIZQ
block LLAVEDER
  ;

guardstmt
  : GUARD expr ELSE LLAVEIZQ block LLAVEDER
  ;
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

```
 breakstmt
   : BREAK
   ;

continuestmt
   : CONTINUE
   ;

returnstmt
   : RETURN expr
   | RETURN
   ;

fnArray
   : ID PUNTO APPEND PARIZQ expr PARDER
   | ID PUNTO REMOVE PARIZQ AT D_PTS expr PARDER
   | ID PUNTO REMOVELAST PARIZQ PARDER
   ;

structCreation
   : STRUCT ID LLAVEIZQ listStructDec LLAVEDER
   ;

listStructDec
   : list=listStructDec (COMA)? (VAR|LET) ID D_PTS types
   | list=listStructDec (COMA)? (VAR|LET) ID D_PTS ID
   | list=listStructDec (COMA)? (VAR|LET) ID D_PTS types IG expr
   | (VAR|LET) ID D_PTS types
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

```
  | (VAR|LET) ID D_PTS ID
  | { $l = []interface{}{} }
  ;

listStructExp
  : list=listStructExp COMA ID D_PTS expr
  | ID D_PTS expr
  | { $l = []interface{}{} }
  ;

listAccessStruct
  : listAccessStruct PUNTO ID
  | ID
  ;

liststmt
  : liststmt COMA expr
  | expr
  | { $l = []interface{}{} }
  ;

listAccessArray
  : listAccessArray CORIZQ expr CORDER
  | CORIZQ expr CORDER
  ;

listArray
  : listArray CORIZQ expr CORDER
  | listArray PUNTO ID
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

```
  | listArray types IG CORIZQ expr CORDER
  | ID
  ;

exprComa
  : exprComa COMA expr
  | expr
  ;

types
  : INT
  | FLOAT
  | STR
  | BOOL
  | CORIZQ types CORDER
  | COMILLA STR COMILLA
  | NIL
  | STRUCT
  | ID
  ;

expr
  : SUB expr
  | expr (SUB_IG|SUM_IG) expr
  | expr (MUL|DIV|MOD) expr
  | expr (ADD|SUB) expr
  | expr (MAY_IG|MAYOR) expr
  | expr (MEN_IG|MENOR) expr
  | expr (IG_IG|DIF) expr
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

```
  | expr AND expr
  | expr OR expr
  | NOT expr
  | PARIZQ expr PARDER
  | types PARIZQ expr PARDER
  | ID PARIZQ listStructExp PARDER
  | CORIZQ CORDER
  | listArray
  | CORIZQ liststmt CORDER
  | NUMBER
  | STRING
  | TRU
  | FAL
  | ID PUNTO COUNT
  | ID PUNTO ISEMPTY
  | NIL
  ;

parametro
  : ID D_PTS types
  | ID D_PTS INOUT types
  | (GUIONBAJO|ID) ID D_PTS types
  | (GUIONBAJO|ID) ID D_PTS INOUT types
  ;

callExp
  : ID PARIZQ listParamsCall PARDER
  ;
```

PROYECTO 1

GRAMÁTICA

PRODUCCIONES

listParamsCall
  : listParamsCall COMA expr
  | expr
  ;