Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de lenguajes y compiladores 1

Gramatica – Proyecto 2

Nombre: Alberto Gabriel Reyes Ning

Sección: C

Carné: 201612174

Gramatica

ini-> main_statements EOF ;

main_statements-> main_statements main_statement

| main_statement ;

main_statement-> run_st END_SENTENCE

| function

| method ;

standard_statements-> standard_statements standard_statement

| standard_statement ;

standard_statement-> declare_array_1 END_SENTENCE

| declare_array_2 END_SENTENCE

| declaration END_SENTENCE

| assign END_SENTENCE

| print_st END_SENTENCE

| println_st END_SENTENCE

| if

| while

| do_while

| do_until

| for

| switch

| call END_SENTENCE

| increment END_SENTENCE

| decrement END_SENTENCE

| BREAK END_SENTENCE

| CONTINUE END_SENTENCE

| RETURN expr END_SENTENCE

| RETURN END_SENTENCE ;

expr-> arithmetic

| relational

| logical

| ternary

| group

| value

| cast

| increment

| decrement

| call

| access_array

| access_matrix

| to_lower_st

| to_upper_st

| round_st

| typeof_st

| tostring_st ;

relational-> expr LESS expr

| expr GREATER expr

| expr LESS_EQUAL expr

| expr GREATER_EQUAL expr

| expr EQUAL expr

| expr NOT_EQUAL expr ;

arithmetic-> expr ADD expr

| expr MINUS expr

| expr PRODUCT expr

| expr DIVISION expr

| expr MODULE expr

| expr POWER expr

| MINUS expr ;

logical-> expr AND expr

| expr OR expr

| NOT expr ;

value-> DECIMAL

| INTEGER

| LOGICAL

| STRING

| CHAR

| IDENTIFIER ;

ternary-> expr TERNARY_IF expr TERNARY_ELSE expr ;

group-> OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

cast-> OPEN_PARENTHESIS TYPE CLOSE_PARENTHESIS expr ;

increment-> IDENTIFIER INCREMENT ;

decrement-> IDENTIFIER DECREMENT ;

list_identifiers-> list_identifiers COMMA IDENTIFIER
| IDENTIFIER ;

declaration-> TYPE list_identifiers
| TYPE list_identifiers ASSIGNMENT expr ;

assign-> list_identifiers ASSIGNMENT expr ;

if-> IF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE

| IF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE ELSE OPEN_BRACE standard_statements CLOSE_BRACE

| IF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE elifs

| IF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE elifs ELSE OPEN_BRACE standard_statements CLOSE_BRACE ;

elifs-> elifs ELIF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE

| ELIF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE ;

while-> WHILE OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE ;

do_while-> DO OPEN_BRACE standard_statements CLOSE_BRACE WHILE OPEN_PARENTHESIS expr CLOSE_PARENTHESIS END_SENTENCE ;

do_until-> DO OPEN_BRACE standard_statements CLOSE_BRACE UNTIL OPEN_PARENTHESIS expr CLOSE_PARENTHESIS END_SENTENCE ;

parameters-> parameters COMMA TYPE IDENTIFIER

| TYPE IDENTIFIER ;

function-> IDENTIFIER OPEN_PARENTHESIS parameters CLOSE_PARENTHESIS TERNARY_ELSE TYPE OPEN_BRACE standard_statements CLOSE_BRACE

| IDENTIFIER OPEN_PARENTHESIS CLOSE_PARENTHESIS TERNARY_ELSE TYPE OPEN_BRACE standard_statements CLOSE_BRACE ;

method-> IDENTIFIER OPEN_PARENTHESIS parameters CLOSE_PARENTHESIS TERNARY_ELSE VOID OPEN_BRACE standard_statements CLOSE_BRACE

| IDENTIFIER OPEN_PARENTHESIS parameters CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE

| IDENTIFIER OPEN_PARENTHESIS CLOSE_PARENTHESIS TERNARY_ELSE VOID OPEN_BRACE standard_statements CLOSE_BRACE

| IDENTIFIER OPEN_PARENTHESIS CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE ;

arguments-> arguments COMMA expr

| expr ;

call-> IDENTIFIER OPEN_PARENTHESIS arguments CLOSE_PARENTHESIS

| IDENTIFIER OPEN_PARENTHESIS CLOSE_PARENTHESIS ;

for-> FOR OPEN_PARENTHESIS for_init END_SENTENCE expr END_SENTENCE for_update CLOSE_PARENTHESIS OPEN_BRACE standard_statements CLOSE_BRACE ;

for_init-> assign

| declaration ;

for_update-> assign

| increment

| decrement ;

switch-> SWITCH OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE cases CLOSE_BRACE

| SWITCH OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE cases DEFAULT TERNARY_ELSE standard_statements CLOSE_BRACE

| SWITCH OPEN_PARENTHESIS expr CLOSE_PARENTHESIS OPEN_BRACE DEFAULT TERNARY_ELSE standard_statements CLOSE_BRACE ;

cases-> cases CASE expr TERNARY_ELSE standard_statements

| CASE expr TERNARY_ELSE standard_statements ;

declare_array_1-> TYPE OPEN_BRACKET CLOSE_BRACKET IDENTIFIER ASSIGNMENT NEW TYPE OPEN_BRACKET expr CLOSE_BRACKET

| TYPE OPEN_BRACKET CLOSE_BRACKET IDENTIFIER ASSIGNMENT OPEN_BRACE list_expr CLOSE_BRACE ;

list_expr-> list_expr COMMA expr

| expr ;

list_list_expr-> list_list_expr COMMA OPEN_BRACE list_expr CLOSE_BRACE

| OPEN_BRACE list_expr CLOSE_BRACE ;

declare_array_2-> TYPE OPEN_BRACKET CLOSE_BRACKET OPEN_BRACKET CLOSE_BRACKET IDENTIFIER ASSIGNMENT NEW TYPE OPEN_BRACKET expr CLOSE_BRACKET OPEN_BRACKET expr CLOSE_BRACKET

| TYPE OPEN_BRACKET CLOSE_BRACKET OPEN_BRACKET CLOSE_BRACKET IDENTIFIER ASSIGNMENT OPEN_BRACE list_list_expr CLOSE_BRACE ;

access_array-> IDENTIFIER OPEN_BRACKET expr CLOSE_BRACKET ;

access_matrix-> IDENTIFIER OPEN_BRACKET expr CLOSE_BRACKET OPEN_BRACKET expr CLOSE_BRACKET ;

print_st-> PRINT OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

println_st-> PRINTLN OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

to_lower_st-> TOLOWER OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

to_upper_st-> TOUPPER OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

round_st-> ROUND OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

typeof_st-> TYPEOF OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

tostring_st-> TOSTRING OPEN_PARENTHESIS expr CLOSE_PARENTHESIS ;

run_st-> RUN IDENTIFIER OPEN_PARENTHESIS list_expr CLOSE_PARENTHESIS

| RUN IDENTIFIER OPEN_PARENTHESIS CLOSE_PARENTHESIS ;