program:- class program '{' field\_decl\_multiple method\_decl\_multiple '}'

field\_decl\_multiple:- /\*epsilon\*/ | field\_decl\_multiple field\_decl\_single ‘;'

field\_decl\_single:- type idList

idList:- identifier id\_single | identifier '[' int\_literal ']' id\_single

id\_single:- /\*epsilon\*/ |',' identifier id\_single | ',' identifier '[' int\_literal ']' id\_single

method\_decl\_multiple:- /\*epsilon\*/ | method\_decl\_single method\_decl\_multiple

method\_decl\_single:- type identifier argumentList block

| void identifier argumentList block

argumentList:- '(' ')' |'(' type identifier arg ')'

arg:- /\*epsilon\*/ |',' type identifier arg

block:- '{' var\_decl\_multiple statement\_multiple ‘}'

var\_decl\_multiple:- /\*epsilon\*/ | var\_decl\_single ';' var\_decl\_multiple

var\_decl\_single:- type identifier variableList

variableList:- /\*epsilon\*/|',' identifier variableList

statement\_multiple:- /\*epsilon\*/ | statement\_multiple statement\_single

statement\_single:- location assignment\_opertor expr ';'

| method\_call ';'

| if condition block else\_block

| for identifier assignment\_opertor expr ',' expr block

| return return\_expr ';'

| break ';'

| continue ';'

| block

else\_block:- /\*epsilon\*/ | else block

condition:- '(' expr ')'

return\_expr:- /\*epsilon\*/ | expr

expr:- location | method\_call | literal | arith\_expr | rel\_expr | equal\_expr

| condition\_expr |'-' expr | '!' expr | '(' expr ')'

location:- identifier | identifier '[' expr ‘]'

method\_call:- method\_name '('parameterList')'

| callout '(' string\_literal callout\_arg ‘)’

method\_name:- identifier

parameterList:- /\*epsilon\*/ | expr parameter

parameter:- /\*epsilon\*/ | ',' expr parameter

literal:- int\_literal | char\_literal | bool\_literal

int\_literal:- decimal\_literal | hex\_literal

arith\_expr:- expr '\*' expr | expr '/' expr | expr '%' expr | expr '+' expr | expr '-' expr

rel\_expr:- expr '<' expr | expr '>' expr | expr less\_equal expr

| expr greater\_equal expr

equal\_expr:- expr equal\_equal expr | expr not\_equal expr

condition\_expr:- expr and expr | expr or expr

callout\_arg:- /\*epsilon\*/ | callout\_arg ',' expr | callout\_arg ',' string\_literal