program ->

translation_unit

translation_unit ->

external_declaration

| translation_unit external_declaration

external_declaration ->

function_definition

| declaration

function_definition ->

type_specifier IDENTIFIER LP RP compound_statement (inside brankets)

| type_specifier IDENTIFIER LP para_list RP compound_statement

(变量声明语句)

declaration ->

para_list SEMI (int 有的函数声明不用声明变量名)

| ANNOTATION

para_list ->

| value

| IDENTIFIER

| declaration

| para_list COMMA para_list

(陈列变量，一个或多个)

init_declarator_list ->

init_declarator

| init_declarator_list COMMA(逗号) init_declarator_list

declaration ->

type_specifier

| DEFINE IDENTIFIER value

| type_specifier init_declarator_list

| IDENTIFIER LP para_list RP

| IDENTIFIER LP RP

| IDENTIFIER LP init_declarator_list RP

init_declarator_list -> init_declarator | init_declarator_list COMMA init_declarator_list

（陈列变量名，初始化或不)

init_declarator ->

declarator (变量名们)

| declarator ASSIGNOP(等号) value

declarator ->

IDENTIFIER

| declarator COMMA(逗号) declarator

| declarator LP RP(new对象或者函数 newNode())

compound_statement ->

LC RC

| LC block_item_list RC

block_item_list ->

block_item

| block_item_list block_item

block_item ->

declaration_list

| statement

| expression SEMI

statement ->

selection_statement | iteration_statement | jump_statement | compound_statement

selection_statement ->

IF LP expression RP statement

| IF LP expression RP statement ELSE statement

| SWITCH LP expression RP statement

| labeled_statement

iteration_statement ->

WHILE LP expression RP statement

| DO statement WHILE LP expression RP SEMI

| LP SEMI SEMI RP statement

| FOR LP init_declarator_list SEMI SEMI RP statement

| FOR LP type_specifier init_declarator_list SEMI SEMI RP statement

| FOR LP init_declarator_list SEMI expression SEMI RP statement

| FOR LP type_specifier init_declarator_list SEMI expression SEMI RP statement

| FOR LP init_declarator_list SEMI SEMI expression RP statement

| FOR LP type_specifier init_declarator_list SEMI SEMI expression RP statement

| FOR LP init_declarator_list SEMI expression SEMI expression RP statement

| FOR LP type_specifier init_declarator_list SEMI expression SEMI expression RP statement

jump_statement -> GOTO value SEMI

| CONTINUE SEMI

| BREAK SEMI

| RETURN SEMI

| RETURN expression SEMI

labeled_statement -> IDENTIFIER SEMICOLON statement

| CASE value SEMICOLON statement

| DEFAULT SEMICOLON statement

(谓词类似)

expression ->

basic_expression

| expression COMMA basic_expression

basic_expression ->

value

| IDENTIFIER

| UNARYOP basic_expression

| basic_expression op basic_expression

value ->

IDENTIFIER

| TRUE

| FALSE

| CONSTANT_INT

| CONSTANT_DOUBLE

| SUB CONSTANT_INT

| SUB CONSTANT_DOUBLE

| VALUE("字符串")

type_specifier ->

TYPE

| VOID

| IDENTIFIER '.' IDENTIFIER(this.type)

| FINAL type_specifier

| STATIC type_specifier

| FINAL STATIC type_specifier

| CONST type_specifier

op -> BINARYOP1

| BINARYOP2

| BINARYOP3

| BINARYOP4

| BINARYOP5

| BINARYOP6

| BINARYOP7

| BINARYOP8

| BINARYOP9

| BINARYOP10

| BINARYOP11

| ASSIGNOP