Gramatica - C#

Gramatica - C#

Gramatica - C#

Programa

program -> declarations?

Declarações

declarations -> declaration declarations?

declaration -> class_declaration | function_declaration | variable_declaration | statement

Declaração de Classe

class_declaration -> 'class' identifier '{' declarations? '}'

Declaração de Função

function_declaration -> return_type identifier '(' parameters? ')' block

return_type -> 'void' | type

parameters -> parameter (',' parameter)*

parameter -> type identifier

Declaração de Variável

variable_declaration -> type identifier ('=' expression)? ';'

type -> 'int' | 'float' | 'string' | 'bool' | 'byte' | 'char' | 'decimal' | 'double' | 'long' | 'object' | 'sbyte' | 'short' | 'uint' | 'ulong' | 'ushort'

Gramatica - C#

Instruções

statement -> if_statement | while_statement | for_statement | foreach_statement | expression_statement | block

if_statement -> 'if' '(' expression ')' statement ('else' statement)?

while_statement -> 'while' '(' expression ')' statement

for_statement -> 'for' '(' (variable_declaration | expression_statement | ';') expression? ';' expression? ')' statement

foreach_statement -> 'foreach' '(' type identifier 'in' expression ')' statement

expression_statement -> expression ';'

block -> '{' declarations? '}'

Expressões

expression -> assignment_expression

assignment_expression -> conditional_expression (assignment_operator assignment_expression)?

conditional_expression -> logical_or_expression ('?' expression ':' expression)?

logical_or_expression -> logical_and_expression ('||' logical_and_expression)*

logical_and_expression -> equality_expression ('&&' equality_expression)*

equality_expression -> relational_expression (('==' | '!=') relational_expression)*

relational_expression -> additive_expression (('<' | '>' | '<=' | '>=') additive_expression)*

additive_expression -> multiplicative_expression (('+' | '-') multiplicative_expression)*

multiplicative_expression -> unary_expression (('*' | '/' | '%') unary_expression)*

unary_expression -> ('+' | '-' | '!' | '~') unary_expression | primary_expression

primary_expression -> identifier | literal | '(' expression ')'

Operadores de Atribuição

Gramatica - C#

assignment_operator -> '=' | '+=' | '-=' | '*=' | '/=' | '%='

Identificadores e Literais

identifier -> [a-zA-Z_][a-zA-Z0-9_]*

literal -> int_literal | float_literal | string_literal | bool_literal

int_literal -> [0-9]+

float_literal -> [0-9]+'.'[0-9]+

string_literal -> '"' [^"]* '"'

bool_literal -> 'true' | 'false'