

# Грамматика

Program → { Statement }

Statement → Scope | If | For | While | DoWhile | Function | Expression ; ;

Scope → '{' Program '}'

If → 'if' '(' Expression ')' Scope ElseIfList ElseOpt

ElseIfList → { 'else' 'if' '(' Expression ')' Scope }

ElseOpt → 'else' Scope |  $\epsilon$

Type → 'int' | 'char' | 'double' | 'string'

For → 'for' '(' ForInit ';' Expression ';' Expression ')' Scope

ForInit → Type VarList

VarList → VarDecl { , , VarDecl }

VarDecl → Identifier [ '=' Expression ] [ '[' Expression ']' ]

Identifier → Letter { Letter | Digit | '\_' }

Letter → 'Q' | 'W' | 'E' | 'R' | 'T' | 'Y' | 'U' | 'I' | 'O' | 'P'  
| 'A' | 'S' | 'D' | 'F' | 'G' | 'H' | 'J' | 'K' | 'L'  
| 'Z' | 'X' | 'C' | 'V' | 'B' | 'N' | 'M'  
| 'q' | 'w' | 'e' | 'r' | 't' | 'y' | 'u' | 'i' | 'o' | 'p'  
| 'a' | 's' | 'd' | 'f' | 'g' | 'h' | 'j' | 'k' | 'l'  
| 'z' | 'x' | 'c' | 'v' | 'b' | 'n' | 'm'

Digit → '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

Number → Digit { Digit }

String → " { Letter | Digit } "

While → 'while' '(' Expression ')' Scope

DoWhile → 'do' Scope 'while' '(' Expression ')' ; ;

Function → 'fn' Type Identifier '(' FunctionArgsOpt ')' Scope

FunctionArgsOpt → FunctionArgs |  $\epsilon$

FunctionArgs → Type Identifier [ '[' , , ] { , , Type Identifier [ '[' , , ] } ]

Expression → AssignmentExpr

AssignmentExpr → EqualityExpr AssignmentTail

AssignmentTail → '=' AssignmentExpr |  $\epsilon$

EqualityExpr → ExprLevel1

EqualityOperator → '==' | '!='

ExprLevel1 → ExprLevel2 [ EqualityOperator ExprLevel2 ]

ComparisonOperator → '<' | '<=' | '>' | '>='

ExprLevel2 → ExprLevel3 [ ComparisonOperator ExprLevel3 ]

ExprLevel3 → ExprLevel4 { ('+' | '-') ExprLevel4 }

ExprLevel4 → ExprLevel5 { ('\*' | '/' | '%') ExprLevel5 }

ExprLevel5 → Primary

Primary → Number | String | Identifier PrimaryTail | '(' Expression ')'

PrimaryTail → '(' CallArgsOpt ')' | '[' Expression ']' PrimaryTail |  $\epsilon$

CallArgsOpt → CallArgs |  $\epsilon$

CallArgs → Expression { , , Expression }