<name>::=<letter>|<name><letter>|<name><digit>

<sign 1> ::= `=` | `+=` | `-=` | `\*=` | `/=` | `%=` | `|=` | `&=` | `^=`

<sign 2> ::= `||`

<sign 3> ::= `&&`

<sign 4> ::= `|`

<sign 4.5> ::= `^`

<sign 5> ::= `&`

<sign 6> ::= `==` | `!=`

<sign 7> ::= `<` | `<=` | `>` | `>=`

<sign 7.5> ::= `<<` | `>>`

<sign 8> ::= `+` | `-`

<sign 9> ::= `\*` | `/` | `%` | `//`

<sign 10> ::= `\*\*`

<sign 11> ::= `++` | `--` | `+` | `-` | `!` | `~`

<sign 12> ::= `++` | `--` // postfix

<sign 13> ::= `.` // member access

<term> ::= <name> [<sign 1> <term 2>] | <name>`[` <term> `]` [ <sign 1> <term 2> ]

<term 2> ::= <term 3> { <sign 2> <term 3> }

<term 3> ::= <term 4> { <sign 3> <term 4> }

<term 4> ::= <term 4.5> { <sign 4> <term 4.5> }

<term 4.5> ::= <term 5> {<sign 4.5> <term 5> }

<term 5> ::= <term 6> { <sign 5> <term 6> }

<term 6> ::= <term 7> { <sign 6> <term 7> }

<term 7> ::= <term 7.5> { <sign 7> <term 7.5> }

<term 7.5> ::= <term 8> {<sign7.5> < term 8>}

<term 8> ::= <term 9> { <sign 8> <term 9> }

<term 9> ::= <term 10> { <sign 9> <term 10> }

<term 10> ::= <term 11> { <sign 10> <term 11> }

<term 11> ::= [ <sign 11> ] <term 12>

<term 12> ::= <term 13> [ <sign 12> ]

<term 13> ::= <term 14> { <sign 13> <method> `(` [ <term> { `,` <term> } ] `)` }

<term 14> ::= <name> | <constant> | <function call> | <term14> `[` <term> `]`

program ::= {<function>} `fn int main()` '{' <comp operator> 'return' [ <term> ] ';' '}'

function ::= `fn` <type> <name> '(' [ <description func> {',' <description func> } ] ')' '{' <comp operator> 'return' [ <term> ] ';' ‘}'

<description func> ::= <type> <name>

<function call> ::= <name> '(' [ <term>, { ',' <term> } ] ‘)'

<comp operator> ::= '{' <operator> { <operator> } '}'

<operator> ::= <comp operator> | <term> ';' | <spec operator> | <description>

<description> ::= <type> <section> { ',' <section> } ‘;'

<section> ::= <name> | <name> = <term>

<spec operator> ::= <operator for> | <operator while> | <operator if>

<operator if> ::= 'if' '(' <term> ')' <comp operator> [ <operator else> ]

<operator for> ::= 'for' '(' <term> ';' <term> ';' <term> ')' <comp operator> |

'for' '(' <description> ';' <term> ';' <term> ')' <comp operator>

<operator while> ::= 'while' '(' <term> ')' <comp operator>

<operator else> ::= 'else' <comp operator>

<type> ::= `int` | `double` | `string` | `vector<` <type> `>`

<method> - это список методов. Он в разработке, мы еще думаем, какие методы будем реализовывать.