translaTOR

<P> ::= <D> {<D>} ⊥

<D> ::= <TYPE> ident <D1>

<D1> ::= ‘(‘ <PARAMSEQ> ‘)’ <S> | {‘,’ ident} ‘;’

<PARAMSEQ> ::= [‘nothing’ | ‘int’ | ‘bool’ | ‘string’ | ‘double’] <IDENT> {‘,’ [‘nothing’ | ‘int’ | ‘bool’ | ‘string’ | ‘double’] <IDENT>} | eps

<S> ::= ‘{‘ <BLOCK>|

‘if’ <IF> | ‘witch’ <WITCH>|

‘for’ <FOR>|

‘while’ <ES> | ‘do’ <DO> |

‘break’ <DELIM> | ‘continue’ <DELIM> | ‘return’ <RETURN>|

[‘nothing’ | ‘int’ | ‘bool’ | ‘string’ | ‘double’] <IDENTSEQ><DELIM>| ‘;’ | <E> <DELIM>

<BLOCK> ::= {<S>}’}’

<IF> ::= <ES> {‘elf’ <ES> } [‘else’ <S> | eps]

<ES> ::= ‘(‘ <E> ‘)’ <S>

<WITCH> ::= ‘{‘ { <E>’:’ <S>} ‘}’

<FOR> ::= ‘(‘ [<E>|eps] ‘;’ <E> ‘;’ [<E> | eps] ‘)’ <S>

<DO> ::= <S> ‘while’ ‘(‘ <E> ‘)’ <DELIM>

<RETURN> ::= [<E> | eps] ‘;’

<DELIM> ::= ‘;’

<IDENTSEQ> ::= ident {‘,’ ident}

<E> ::= <E1>

<ESEQ> ::= <E2> {‘,’<E2>} // пунктуационная запятая

<E1> ::= <E2> {‘,’<E2>} // оператор запятая – check\_bin

<E2> ::= <E3> {[‘=’ | ‘+=’ | ‘-=’ | ‘\*=’ | ‘/=’ | ‘%=’ | ‘&=’ | ‘|=’ | ‘<<=’ | ‘>>=’] <E3>} – check\_bin

<E3> ::= <E4> {‘||’<E4>} – check\_bin

<E4> ::= <E5> {‘&&’<E5>} – check\_bin

<E5> ::= <E6> {‘|’<E6>} – check\_bin

<E6> ::= <E7> {‘^’<E7>} – check\_bin

<E7> ::= <E8> {‘&’<E8>} – check\_bin

<E8> ::= <E9> {[‘==’ | ‘!=’]<E9>} – check\_bin

<E9> ::= <E10> {[‘<’ | ‘>’ | ‘<=’ | ‘>=’]<E10>} – check\_bin

<E10> ::= <E11> {[‘<<’ | ‘>>’]<E11>} – check\_bin

<E11> ::= <E12> {[‘+’ | ‘-‘]<E12>} – check\_bin

<E12> ::= <E13> {[‘\*’ | ‘/’ | ‘%’]<E13>} – check\_bin

<E13> ::= {[‘++’ | ‘—‘ | ‘+’ | ‘-‘ | ‘!’ | ‘~’ | ‘sizeof’]}<E14> -- check\_uno

<E14> ::= <E15>{‘[‘ <E> ‘]’ | ‘(‘ [<ESEQ> | eps] ‘)’| ‘++’ | ‘—‘} –check\_braces, check\_postfix

<E15> ::= ‘(‘ <E> ‘)’ | ident | ‘comein’ | ‘goout’ |

string\_literal | int\_literal | double\_literal | boolean\_literal

<TYPE> ::= ‘nothing’ | ‘int’ | ‘bool’ | ‘string’ | ‘double’

Хранить is\_lvalue

Comein и gout заменять на объекты типа IOSTREAMS

String[i] – это тоже string