

```

    program ::= class id { variable_declarations method_declarations }

variable_declarations ::= type variable_list ; variable_declarations |  $\epsilon$ 

    type ::= int | real

variable_list ::= variable variable_list2

variable_list2 ::= , variable variable_list2 |  $\epsilon$ 

    variable ::= id variable2

variable2 ::= [ num ] |  $\epsilon$ 

method_declarations ::= method_declaration more_method_declarations

more_method_declarations ::= method_declaration more_method_declarations |  $\epsilon$ 

    method_declaration ::= static method_return_type id ( parameters )
    { variable_declarations statement_list }

method_return_type ::= type | void

    parameters ::= parameter_list |  $\epsilon$ 

parameter_list ::= type id parameter_list2

parameter_list2 ::= , type id parameter_list2 |  $\epsilon$ 

    statement_list ::= statement statement_list |  $\epsilon$ 

id_starting_statement ::= id rest_of_id_starting_statement ;

rest_of_id_starting_statement ::= ( expression_list )
    | incdecop
    | = expression
    | [ expression ] = expression

    statement ::= id_starting_statement
    | if ( expression ) statement_block optional_else
    | for ( variable_loc = expression ; expression ; variable_loc
    | incdecop ) statement_block
    | return optional_expression ;
    | break ;
    | continue ;
    | statement_block

optional_expression ::= expression |  $\epsilon$ 

    statement_block ::= { statement_list }

    optional_else ::= else statement_block |  $\epsilon$ 

```

$$\begin{aligned}
expression_list &::= expression \ more_expressions \mid \epsilon \\
more_expressions &::= , \ expression \ more_expressions \mid \epsilon \\
expression &::= simple_expression \ expression2 \\
expression2 &::= \mathbf{relop} \ simple_expression \mid \epsilon \\
simple_expression &::= term \ simple_expression2 \mid sign \ term \ simple_expression2 \\
simple_expression2 &::= \mathbf{addop} \ term \ simple_expression2 \mid \epsilon \\
term &::= factor \ term2 \\
term2 &::= \mathbf{mulop} \ factor \ term2 \mid \epsilon \\
id_starting_factor &::= \mathbf{id} \ rest_of_id_starting_factor \\
rest_of_id_starting_factor &::= \begin{array}{l} (\ expression_list \) \\ \mid [\ expression \] \\ \mid \epsilon \end{array} \\
factor &::= id_starting_factor \mid \mathbf{num} \mid (\ expression \) \mid ! \ factor \\
variable_loc &::= \mathbf{id} \ variable_loc_rest \\
variable_loc_rest &::= [\ expression \] \mid \epsilon \\
sign &::= + \mid -
\end{aligned}$$