program ::= **class** **id** { *variable_declarations* *method_declarations* }

variable_declarations ::= *type* *variable_list* **;** *variable_declarations* | **ε**

type ::= **int** | **real**

variable_list ::= *variable* *more_variables*

more_variables ::= **,** *variable_list* | **ε**

variable ::= **id** *opt_array*

opt_array ::= **[num]** | **ε**

method_declarations ::= *method_declaration* *more_method_declarations*

more_method_declarations ::= *method_declaration* *more_method_declarations* | **ε**

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

method_return_type ::= *type* | **void**

parameters ::= *parameter* *more_parameters* | **ε**

more_parameters ::= **,** *parameters* | **ε**

parameter ::= *type* **id**

statement_list ::= *statement* *statement_list* | **ε**

statement ::= *variable_loc* **=** *expression* **;**
| **id** (*expression_list*) **;**
| **if** **(** *expression* **)** *statement_block* *optional_else*
| **for** **(** *variable_loc* = *expression* **;** *expression* **;** *incr_decr_var* **)** *statement_block*
| **return** *optional_expression* **;**
| **break** **;**
| **continue** **;**
| *incr_decr_var* **;**
| *statement_block*

optional_expression ::= *expression* | **ε**

statement_block ::= { *statement_list* }

incr_decr_var ::= *variable_loc* **incdecop**

optional_else ::= **else** *statement_block* | ε

expression_list ::= *expression* *more_expressions* | **ε**

more_expressions ::= **,** *expression* *more_expressions* | **ε**

expression ::= *simple_expression* *optional_relop*

optional_relop ::= **relop** *simple_expression* | **ε**

simple_expression ::= *sign* *term* *optional_addops* | *term* *optional_addops*

optional_addops ::= **addop** *term* *optional_addops* | **ε**

term ::= *factor* *optional_mulop*

optional_mulop ::= **mulop** *term* | **ε**

factor ::= variable_loc | **id** **(** *expression_list* **)** | **num** | **(** *expression* **)** | **!** *factor*

variable_loc ::= **id** *opt_index*

opt_index ::= **[** *expression* **]** | **ε**

sign ::= + | -

FIRST(sign) = FIRST(+) u FIRST(-) = { +,- }

FIRST(type) = FIRST(int) u FIRST(real) = { int, real }

FIRST(opt_index) = FIRST([) u FIRST(ε) = { [, ε }

FIRST(variable_loc) = FIRST(id) = { id }

FIRST(factor) = FIRST(variable_loc) u FIRST(id) u FIRST(num) u FIRST(!) u FIRST(() = { id, num, (, ! }

FIRST(optional_mulop) = FIRST(mulop) u FIRST(ε) = { mulop, ε }

FIRST(term) = FIRST(factor) = { id, num, (, !}

FIRST(optional_addops) = FIRST(addop) u FIRST(ε) = { addop, ε }

FIRST(simple_expression) = FIRST(sign) u FIRST(term) = { +, -, id, num, (, ! }

FIRST(optinal_relop) = FIRST(relop) u FIRST(ε) = { relop, ε }

FIRST(expression) = FIRST(simple_expression) = { +, -, id, num, (, ! }

FIRST(more_expressions) = FIRST(,) u FIRST(ε) = { "," , ε}

FIRST(expression_list) = FIRST(expression) u FIRST(ε) = { +, -, id, num, (, !, ε}

FIRST(optional_else) = FIRST(else) u FIRST(ε) = { else, ε }

FIRST(incr_decr_var) = FIRST(variable_loc) = { id }

FIRST(statement_block) = FIRST({) = { "{" }

FIRST(optional_expression) = FIRST(expression) u FIRST(ε) = { +, -, id, num, (, ! , ε}

FIRST(statement) = FIRST(variable_loc) u FIRST(id) u FIRST(if) u FIRST(for) u FIRST(return) u FIRST(break) u FIRST(continue) u FIRST(incr_decr_var) u FIRST(statement_block) = { id, if, for, return, break, continue, "{" }

FIRST(statement_list) = FIRST(statement) u FIRST(ε) = { id, if, for, return, break, continue, ε, "{" }

FIRST(more_parameters) = FIRST(,) u FIRST(ε) = {"," , ε}

FIRST(parameter_list) = FIRST(type) = { int, real }

FIRST(parameters) = FIRST(parameter_list) u FIRST(ε) = { int, real, ε}

FIRST(method_return_type) = FIRST(type) u FIRST(void) = { int, real, void}

FIRST(method_declaration) = FIRST(static) = { static }

FIRST(more_method_declarations) = FIRST(method_declaration) u FIRST(ε) = { static, ε }

FIRST(method_declarations) = FIRST(method_declaration) = { static }

FIRST(opt_array) = FIRST([) u FIRST(ε) = { [, ε }

FIRST(variable) = FIRST(id) = { id }

FIRST(more_variables) = FIRST(,) u FIRST(ε) = { "," , ε }

FIRST(variable_list) = FIRST(variable) = { id }

FIRST(variable_declarations) = FIRST(type) u FIRST(ε) = { int, real, ε }

FIRST(program) = FIRST(class) = { class }