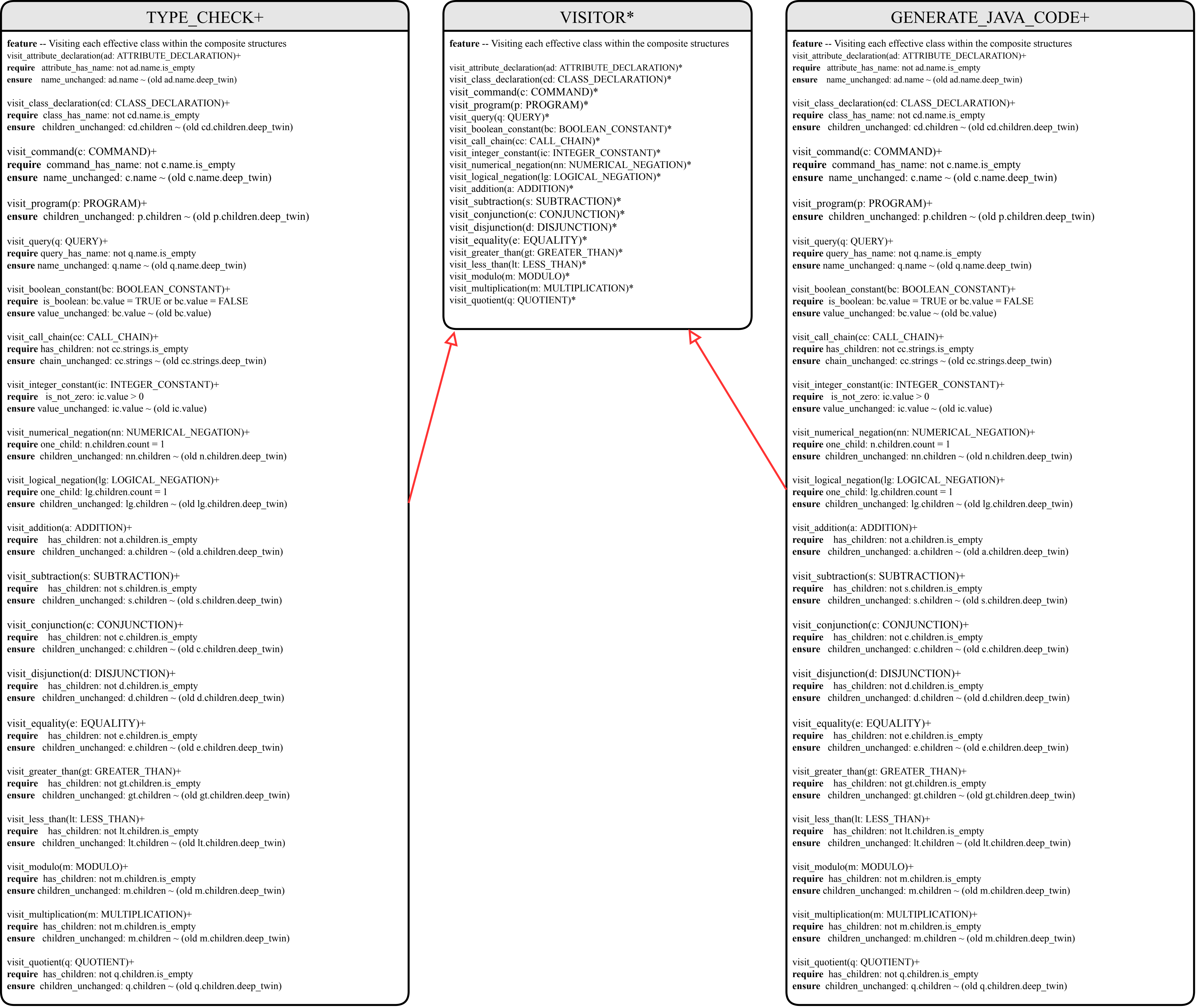

TYPE_CHECK+

feature -- Visiting each effective class within the composite structures
visit_attribute_declaration(ad: ATTRIBUTE_DECLARATION)+
require attribute_has_name: not ad.name.is_empty
ensure name_unchanged: ad.name ~ (old ad.name.deep_twin)

visit_class_declaration(cd: CLASS_DECLARATION)+
require class_has_name: not cd.name.is_empty
ensure children_unchanged: cd.children ~ (old cd.children.deep_twin)

visit_command(c: COMMAND)+
require command_has_name: not c.name.is_empty
ensure name_unchanged: c.name ~ (old c.name.deep_twin)

visit_program(p: PROGRAM)+
ensure children_unchanged: p.children ~ (old p.children.deep_twin)

visit_query(q: QUERY)+
require query_has_name: not q.name.is_empty
ensure name_unchanged: q.name ~ (old q.name.deep_twin)

visit_boolean_constant(bc: BOOLEAN_CONSTANT)+
require is_boolean: bc.value = TRUE or bc.value = FALSE
ensure value_unchanged: bc.value ~ (old bc.value)

visit_call_chain(cc: CALL_CHAIN)+
require has_children: not cc.strings.is_empty
ensure chain_unchanged: cc.strings ~ (old cc.strings.deep_twin)

visit_integer_constant(ic: INTEGER_CONSTANT)+
require is_not_zero: ic.value > 0
ensure value_unchanged: ic.value ~ (old ic.value)

visit_numerical_negation(nn: NUMERICAL_NEGATION)+
require one_child: n.children.count = 1
ensure children_unchanged: nn.children ~ (old n.children.deep_twin)

visit_logical_negation(lg: LOGICAL_NEGATION)+
require one_child: lg.children.count = 1
ensure children_unchanged: lg.children ~ (old lg.children.deep_twin)

visit_addition(a: ADDITION)+
require has_children: not a.children.is_empty
ensure children_unchanged: a.children ~ (old a.children.deep_twin)

visit_subtraction(s: SUBTRACTION)+
require has_children: not s.children.is_empty
ensure children_unchanged: s.children ~ (old s.children.deep_twin)

visit_conjunction(c: CONJUNCTION)+
require has_children: not c.children.is_empty
ensure children_unchanged: c.children ~ (old c.children.deep_twin)

visit_disjunction(d: DISJUNCTION)+
require has_children: not d.children.is_empty
ensure children_unchanged: d.children ~ (old d.children.deep_twin)

visit_equality(e: EQUALITY)+
require has_children: not e.children.is_empty
ensure children_unchanged: e.children ~ (old e.children.deep_twin)

visit_greater_than(gt: GREATER_THAN)+
require has_children: not gt.children.is_empty
ensure children_unchanged: gt.children ~ (old gt.children.deep_twin)

visit_less_than(lt: LESS_THAN)+
require has_children: not lt.children.is_empty
ensure children_unchanged: lt.children ~ (old lt.children.deep_twin)

visit_modulo(m: MODULO)+
require has_children: not m.children.is_empty
ensure children_unchanged: m.children ~ (old m.children.deep_twin)

visit_multiplication(m: MULTIPLICATION)+
require has_children: not m.children.is_empty
ensure children_unchanged: m.children ~ (old m.children.deep_twin)

visit_quotient(q: QUOTIENT)+
require has_children: not q.children.is_empty
ensure children_unchanged: q.children ~ (old q.children.deep_twin)

VISITOR*

feature -- Visiting each effective class within the composite structures

visit_attribute_declaration(ad: ATTRIBUTE_DECLARATION)*
visit_class_declaration(cd: CLASS_DECLARATION)*
visit_command(c: COMMAND)*
visit_program(p: PROGRAM)*
visit_query(q: QUERY)*
visit_boolean_constant(bc: BOOLEAN_CONSTANT)*
visit_call_chain(cc: CALL_CHAIN)*
visit_integer_constant(ic: INTEGER_CONSTANT)*
visit_numerical_negation(nn: NUMERICAL_NEGATION)*
visit_logical_negation(lg: LOGICAL_NEGATION)*
visit_addition(a: ADDITION)*
visit_subtraction(s: SUBTRACTION)*
visit_conjunction(c: CONJUNCTION)*
visit_disjunction(d: DISJUNCTION)*
visit_equality(e: EQUALITY)*
visit_greater_than(gt: GREATER_THAN)*
visit_less_than(lt: LESS_THAN)*
visit_modulo(m: MODULO)*
visit_multiplication(m: MULTIPLICATION)*
visit_quotient(q: QUOTIENT)*

GENERATE_JAVA_CODE+

feature -- Visiting each effective class within the composite structures
visit_attribute_declaration(ad: ATTRIBUTE_DECLARATION)+
require attribute_has_name: not ad.name.is_empty
ensure name_unchanged: ad.name ~ (old ad.name.deep_twin)

visit_class_declaration(cd: CLASS_DECLARATION)+
require class_has_name: not cd.name.is_empty
ensure children_unchanged: cd.children ~ (old cd.children.deep_twin)

visit_command(c: COMMAND)+
require command_has_name: not c.name.is_empty
ensure name_unchanged: c.name ~ (old c.name.deep_twin)

visit_program(p: PROGRAM)+
ensure children_unchanged: p.children ~ (old p.children.deep_twin)

visit_query(q: QUERY)+
require query_has_name: not q.name.is_empty
ensure name_unchanged: q.name ~ (old q.name.deep_twin)

visit_boolean_constant(bc: BOOLEAN_CONSTANT)+
require is_boolean: bc.value = TRUE or bc.value = FALSE
ensure value_unchanged: bc.value ~ (old bc.value)

visit_call_chain(cc: CALL_CHAIN)+
require has_children: not cc.strings.is_empty
ensure chain_unchanged: cc.strings ~ (old cc.strings.deep_twin)

visit_integer_constant(ic: INTEGER_CONSTANT)+
require is_not_zero: ic.value > 0
ensure value_unchanged: ic.value ~ (old ic.value)

visit_numerical_negation(nn: NUMERICAL_NEGATION)+
require one_child: n.children.count = 1
ensure children_unchanged: nn.children ~ (old n.children.deep_twin)

visit_logical_negation(lg: LOGICAL_NEGATION)+
require one_child: lg.children.count = 1
ensure children_unchanged: lg.children ~ (old lg.children.deep_twin)

visit_addition(a: ADDITION)+
require has_children: not a.children.is_empty
ensure children_unchanged: a.children ~ (old a.children.deep_twin)

visit_subtraction(s: SUBTRACTION)+
require has_children: not s.children.is_empty
ensure children_unchanged: s.children ~ (old s.children.deep_twin)

visit_conjunction(c: CONJUNCTION)+
require has_children: not c.children.is_empty
ensure children_unchanged: c.children ~ (old c.children.deep_twin)

visit_disjunction(d: DISJUNCTION)+
require has_children: not d.children.is_empty
ensure children_unchanged: d.children ~ (old d.children.deep_twin)

visit_equality(e: EQUALITY)+
require has_children: not e.children.is_empty
ensure children_unchanged: e.children ~ (old e.children.deep_twin)

visit_greater_than(gt: GREATER_THAN)+
require has_children: not gt.children.is_empty
ensure children_unchanged: gt.children ~ (old gt.children.deep_twin)

visit_less_than(lt: LESS_THAN)+
require has_children: not lt.children.is_empty
ensure children_unchanged: lt.children ~ (old lt.children.deep_twin)

visit_modulo(m: MODULO)+
require has_children: not m.children.is_empty
ensure children_unchanged: m.children ~ (old m.children.deep_twin)

visit_multiplication(m: MULTIPLICATION)+
require has_children: not m.children.is_empty
ensure children_unchanged: m.children ~ (old m.children.deep_twin)

visit_quotient(q: QUOTIENT)+
require has_children: not q.children.is_empty
ensure children_unchanged: q.children ~ (old q.children.deep_twin)