

LANGUAGE_COMMANDS+

feature -- model attributes
 classes: **LINKED_LIST**[**CLASS_INSTANCE**]
 current_exp: **EXPRESSION**
 pretty: **PRETTY_PRINTING**

feature {**NONE**} -- Initialization
 make

feature -- model operations
 reset+
 add_class (name: **STRING**) +
 require
 not_assigning: \neg assign_in_progress
 not_existing_class: \neg existing_class (name)
 add_attr (cn, fn, type: **STRING**) +
 require
 not_assigning: \neg assign_in_progress
 existing_class: existing_class (cn)
 not_existing_feature: \neg existing_fn (cn, fn)
 valid_return_type: valid_ref_type (type)
 add_command (cn, name: **STRING**; params: **ARRAY**[**TUPLE**[**STRING**, **STRING**]]) +
 require
 not_assigning: \neg assign_in_progress
 existing_class: existing_class (cn)
 not_existing_feature: \neg existing_fn (cn, name)
 not_clashing_params: \neg params_clash_with_classes (params)
 not_duplicated_params: \neg params_are_duplicated (params)
 params_are_valid_ref_type: params_type_valid (params)
 add_query (cn, name: **STRING**; params: **ARRAY**[**TUPLE**[**STRING**, **STRING**]]; type: **STRING**) +
 require
 not_assigning: \neg assign_in_progress
 existing_class: existing_class (cn)
 not_existing_feature: \neg existing_fn (cn, name)
 not_clashing_params: \neg params_clash_with_classes (params)
 not_duplicated_params: \neg params_are_duplicated (params)
 params_are_valid_ref_type: params_type_valid (params)
 valid_return_type: valid_ref_type (type)
 add_assignment (cn, fn, n: **STRING**) +
 require
 not_assigning: \neg assign_in_progress
 existing_class: existing_class (cn)
 feature_exists: find_feature (cn, fn) > 0
 feature_is_not_attr: \neg existing_attr (cn, fn)
 add_int_expression (v: **STRING**) +
 require
 assign_in_progress: assign_in_progress
 add_callchain_expression (v: **ARRAY**[**STRING**]) +
 require
 assign_in_progress: assign_in_progress
 chain_not_empty: \neg v.is_empty
 generate_java_code+
 type_check+
 require
 not_assigning: \neg assign_in_progress

feature -- queries
 out: **STRING**

classes+: **LINKED_LIST**[...]

commands+: **LINKED_LIST**[...]

queries+: **LINKED_LIST**[...]

expressions+: **LINKED_LIST**[...]

EXPRESSION*

feature -- Attributes
 value+, type: **STRING**
 filled+: **BOOLEAN**

feature -- Initialize
 init+

feature -- Accept a visitor for operation
 accept (v: **VISITOR**)*

feature -- Setters
 set_filled (b: **BOOLEAN**) +

COMPOSITE[G]*

feature -- Attributes
 exp+: **G**

feature -- Setters
 set_exp (e: **G**) +

COMPOSITE_2[G]*

feature -- Attributes
 left+, right+: **G**

feature -- Setters
 set_left (e: **G**) +
 set_right (e: **G**) +

+
LOGICAL_AND

+
EQUAL

+
ADDITION

+
UNARY_OP

+
NIL

+
CALL_CHAIN

+
BOOLEAN_CONSTANT

+
NEXT

+
BINARY_OP

+
INTEGER_CONSTANT

current_exp+