ARRAYED_CONTAINER

feature -- { NONE }
-- Implementation of an arrayed-container
imp: ARRAY[STRING]
feature -- Commands
assign_at (i: **INTEGER**; s: **STRING**)
-- Change the value at position 'i' to 's'.
require
valid_index: valid_index (i)
ensure
size_unchanged: imp.count = (**old** imp.twin).count
item_assigned: imp[i] ~ s
others_unchanged: ∀j : 1 ≤ j ≤ imp.count : j ≠ i ⇒ imp[j] ~ (**old** imp.twin) [j]
delete_at (i: **INTEGER**)
-- Delete element stored at index 'i'.
require
valid_index: valid_index (i)
ensure
size_unchanged: imp.count = (**old** imp.twin).count - 1
left_half_the same: ∀j : imp.lower ≤ j ≤ i-1 : ⇒ imp[j] ~ (**old** imp.twin) [j]
right_half_the_same: ∀j : i ≤ j ≤ ((**old** imp.twin).upper) - 1 : ⇒ imp[j] ~ (**old** imp.twin) [j + 1]
insert_at (i: **INTEGER;** s: **STRING**)
-- Insert value 's' into index 'i'.
require
valid_index: valid_index (i)
ensure
size_changed: imp.count = (**old** imp.twin).count + 1
left_half_the same: ∀j : imp.lower ≤ j ≤ i-1 : ⇒ imp[j] ~ (**old** imp.twin) [j]
right_half_the_same: ∀k : i ≤ k ≤ (**old** imp.twin).upper : ⇒ imp[k + 1] ~ (**old** imp.twin) [k]
insert_last (s: **STRING**)
-- Insert 's' as the last element of the container.
ensure
size_changed: count = (**old** imp.twin).count + 1
last_inserted: imp[imp.upper] ~ s
others_unchanged: ∀j : imp.lower ≤ j ≤ (**old** imp.twin).upper : ⇒ imp[j] ~ (**old** imp.twin) [j]
remove_first
-- Remove first element from the container.
require
not_empty: count > 0
ensure
size_changed: imp.count = (**old** imp.twin).count - 1
others_unchanged: ∀j : (**old** imp.twin).lower ≤ j ≤ imp.upper : ⇒ imp[j] ~ (**old** imp.twin) [j + 1]
feature -- Constructors
make
-- Initialize an empty container.
ensure
empty_container: imp.count = 0
count: **INTEGER**
-- Gets the number of elements in the array by calculating it using a formula
ensure
Result = imp.upper - imp.lower + 1
get_at (i: **INTEGER**): **STRING**
-- Return the element stored at index 'i'
require
valid_index: valid_index (i)
ensure
size_unchanged: imp.count = (**old** imp.twin).count
result_correct: **Result** = imp[i]
no_elements_changed: ∀j : **old** imp.lower ≤ j ≤ (**old** imp.twin).upper : ⇒ imp[j] ~ (**old** imp.twin) [j]
valid_index (i: **INTEGER**): **BOOLEAN**
-- Is 'i' a valid index of current container?
ensure
size_unchanged: imp.count = (**old** imp.twin).count
result_correct: **Result** = imp[i]
no_elements_changed: ∀j : **old** imp.lower ≤ j ≤ (**old** imp.twin).upper : ⇒ imp[j] ~ (**old** imp.twin) [j]
invariant
consistency: imp.count = count