

# ARRAYED\_CONTAINER

## feature -- Commands

```
assign_at (i: INTEGER_32; s: STRING_8)
  -- 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:  $\forall j: 1 \leq j \leq \text{imp.count} : i \neq j \Rightarrow \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 

delete_at (i: INTEGER_32)
  require
    valid_index: valid_index (i)
  ensure
    size_changed: imp.count  $\neq$  (old imp.twin).count
    left_half_the_same:  $\forall j: 1 \leq j \leq i : i < j \Rightarrow \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 
    right_half_the_same:  $\forall j: 1 \leq j \leq \text{imp.count} : i < j \Rightarrow \text{imp}[j+1] \sim (\text{old imp.twin}) [j]$ 

insert_at (i: INTEGER_32; s: STRING_8)
  -- Insert value 's' into index 'i'.
  require
    valid_index: valid_index (i)
  ensure
    size_changed: imp.count  $\neq$  (old imp.twin).count
    inserted_at_i: imp [i] ~ s
    left_half_the_same:  $\forall j: 1 \leq j \leq i : i < j \Rightarrow \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 
    right_half_the_same:  $\forall j: 1 \leq j \leq \text{imp.count} : i < j \Rightarrow \text{imp}[j+1] \sim (\text{old imp.twin}) [j]$ 

insert_last (s: STRING_8)
  -- Insert 's' as the last element of the container.
  ensure
    size_changed: imp.count  $\neq$  (old imp.twin).count
    last_inserted: imp [count] = s
    others_unchanged:  $\forall j: 1 \leq j \leq \text{imp.count} : i \neq j \Rightarrow \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 

remove_first
  -- Remove first element from the container.
  require
    not_empty: count  $\geq$  1
  ensure
    size_changed: imp.count  $\neq$  (old imp.twin).count
    others_unchanged:  $\forall j: 1 \leq j \leq \text{imp.count} : j \geq \text{imp.lower} \Rightarrow \text{imp}[j] \sim (\text{old imp.twin}) [j+1]$ 
```

## feature -- Constructors

```
make
  -- Initialize an empty container.
  ensure
    empty_container: count = 0
```

## feature -- Queries

```
count: INTEGER_32
  -- Your task
  -- Number of items currently stored in the container.
  -- It is up to you to either implement 'count' as an attribute,
  -- or to implement 'count' as a query (uniform access principle).

get_at (i: INTEGER_32): STRING_8
  -- Return the element stored at index 'i'.
  -- you need to make sure that the value is not larger than max size of array
  require
    valid_index: valid_index (i)
  ensure
    size_unchanged: imp.count = (old imp.twin).count
    result_correct:  $\forall j: 1 \leq j \leq \text{imp.lower} : \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 
    no_elements_changed:  $\forall j: 1 \leq j \leq \text{imp.count} : \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 

valid_index (i: INTEGER_32): BOOLEAN
  -- Is 'i' a valid index of current container?
  ensure
    size_unchanged: imp.count = (old imp.twin).count
    result_correct:  $\forall j: 1 \leq j \leq \text{imp.lower} : \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 
    no_elements_changed:  $\forall j: 1 \leq j \leq \text{imp.count} : \text{imp}[j] \sim (\text{old imp.twin}) [j]$ 

invariant
  consistency: imp.count = count

end -- class ARRAYED_CONTAINER
```