ITERABLE[G]*

feature
new_cursor*: **ITERATION_CURSOR[G]**
--Fresh cursor associated with current structure

new_cursor*

ITERATION_CURSOR[G]*

feature
   item*: G
--Item at the current cursor
position
 feature
   after*: BOOLEAN
 --Are there no more items to
iterate over?
feature
   forth*
-- Move to next position.

DATA_SET[V1, V2, K]+

feature --Attributes
data_item_1: V1
data_item_2: V2
key: K
feature -- Equality
  is_equal+(like Current) : BOOLEAN

REPOSITORY[K, D1, D2]+

feature --Attributes
     keys: LINKED_LIST[K]
     data_items_1: ARRAY[D1]
     data_items_2: HASH_TABLE[D2, K]
feature --Mutators
  check_in (d1: D1; d2: D2; k: K)
  require non_existing_key: ∀j: 1 ≤ j ≤ keys.count:
          keys.i_th(j) ≠ k
  ensure repository_count_incremented:
          keys.count = **old** key.count + 1
  check_out (k: KEY)
  require existing_key: ∃j: 1 ≤ j ≤ keys.count:
    keys.i_th(j) = k
  ensure repository_count_decremented:
          keys.count = **old** key.count - 1
invariant
consistent_keys_data_items_counts:
     keys.count = data_items_1.count ∧
     keys.count = data_items_2.count

new_cursor+

another_cursor+

item+

+
TUPLE_ITERATION_CURSOR[K, D1, D2]

+
DATA_SET_ITERATION_CURSOR[D1, D2, K]

repos+

+
EXAMPLE_REPOSITORY_TESTS