CLIENT

CLIENT_APPLICATION+

class interface
DICTIONARY [V, K]

feature -- Alternative Iteration Cursor
another_cursor: ITERATION_CURSOR [ENTRY [V, K]]
add_entry (v: V; k: K)
-- Add a new entry with key 'k' and value 'v'.
-- It is required that 'k' is not an existing search key in the dictionary.
require
non_existing_key: not exists (k)
ensure
entry_added: (get_value (k) ~ v)

remove_entry (k: K)
-- Remove the corresponding entry whose search key is 'k'.
-- It is required that 'k' is an existing search key in the dictionary.
require
existing_key: exists (k)
ensure
dictionary_count_decremented: count = (old count - 1)
key_removed: not exists (k)
feature -- Constructor
feature -- Feature(s) required by ITERABLE
new_cursor: ITERATION_CURSOR [TUPLE [V, K]]

Dictionary*

cluster_name

ITERABLE+

feature -- { A, B, C }
  -- features exported to classes A, B, and C
feature -- { NONE }
  -- private features
invariant
  inv_1: 0 < balance < 1,000,000

new_cursor*

ITERATION_CURSOR[G]*

after* : BOOLEAN
  --Are there no more item to iterate over
item*: G
  -- Item at current cursor position.
   ? valid_position: not after
forth *
i-- Move to next position.
 ? valid_position: not after

ENTRY_ITERATION_CURSOR[ENTRY[V,K]]

after* : BOOLEAN
  --Are there no more item to iterate over
item*: G
  -- Item at current cursor position.
   ? valid_position: not after
forth *
i-- Move to next position.
 ? valid_position: not after

new_cursor+ *

TUPLE_ITERATION_CURSOR[TUPLE[V,K]]

after* : BOOLEAN
  --Are there no more item to iterate over
item*: G
  -- Item at current cursor position.
   ? valid_position: not after
forth *
i-- Move to next position.
 ? valid_position: not after

new_cursor+ *

+
ENTRY

+
TUPLE

ITERABLE COLLECTION