EECS3311

Fall 2017

Saad Saeed

saeeds28

class interface

DICTIONARY [V, K]

create

make

feature -- Alternative Iteration Cursor

another\_cursor: ITERATION\_CURSOR [ENTRY [V, K]]

feature -- Commands

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: values [values.count] ~ v and keys [keys.count] ~ k

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: values.count = (old values.twin.count) - 1

key\_removed: not exists (k)

feature -- Constructor

make

-- Initialize an empty dictionary.

ensure

empty\_dictionary: (values.count = 0) and (keys.count = 0)

object\_equality\_for\_keys: keys.object\_comparison

object\_equality\_for\_values: values.object\_comparison

feature -- Queries

count: INTEGER\_32

-- Number of entries in the dictionary.

ensure

correct\_result: Result = values.count

exists (k: K): BOOLEAN

-- Does key 'k' exist in the dictionary?

ensure

correct\_result: across

1 |..| keys.count as l

some

keys [l.item] ~ k

end

get\_keys (v: V): ITERABLE [K]

-- Return an iterable collection of keys that are associated with value 'v'.

-- Hint: Refere to the architecture BON diagram of the Iterator Pattern, to see

-- what classes can be used to instantiate objects that are iterable.

ensure

correct\_result: across

Result as c

all

values.at (keys.index\_of (c.item, 1)) ~ v

end

get\_value (k: K): detachable V

-- Return the assocated value of search key 'k' if it exists.

-- Void if 'k' does not exist.

-- Declaring "detachable" besides the return type here indicates that

-- the return value might be void (i.e., null).

ensure

case\_of\_void\_result: keys.has (k) = False implies Result ~ Void

case\_of\_non\_void\_result: True

keys.has (k) = True implies Result /~ Void

feature --Iterable method

new\_cursor: ITERATION\_CURSOR [TUPLE [V, K]]

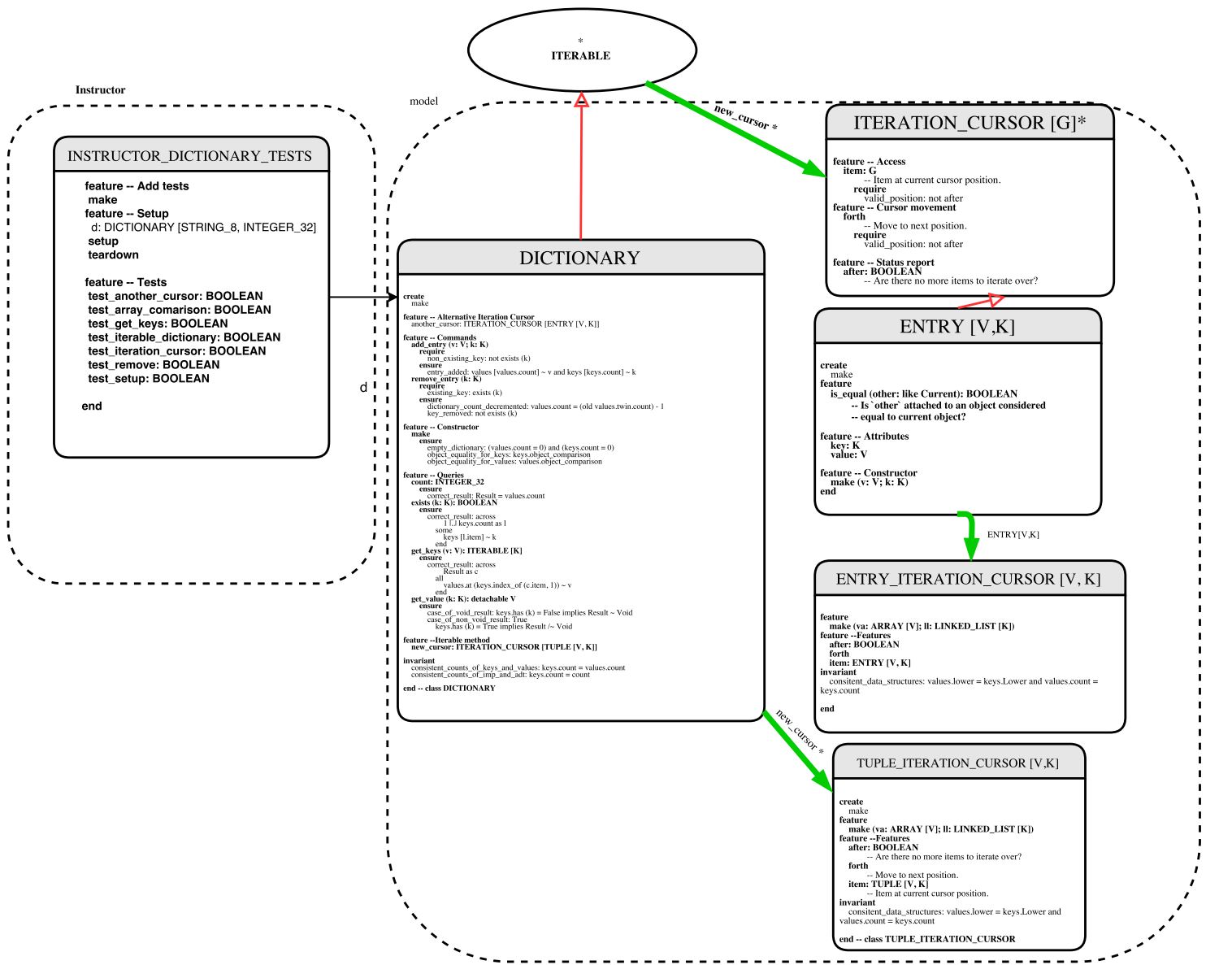
-- Fresh cursor associated with current structure

invariant

consistent\_counts\_of\_keys\_and\_values: keys.count = values.count

consistent\_counts\_of\_imp\_and\_adt: keys.count = count

end --

Architectural Diagram

The iterator pattern is implemented as follows:  
The Dictionary class is made to inherit from the ITERABLE class of type tuple[V,K]. The new\_cursor method that is required in the class is implemented; it returns an ITERATION\_CURSOR[TUPLE[V,K]] object. This object is an instantiation of the TUPLE\_ITERATION\_CURSOR class. This class inherits from ITERATION\_CURSOR of type TUPLE[V,K]. Here the three required features item, forth, and after are defined. On the client side, the client calls the d.new\_cursor method to get a new iterator.

The another\_cursor method returns an ITERATION\_CURSOR[ENTRY[V,K]] object. This object is an instantiation of the ENTRY\_ITERATION\_CURSOR class. This class inherits from ITERATION\_CURSOR of type ENTRY[V,K]. Here the three required features item, forth, and after are defined. Furthermore, the method is\_equal is redefined in the class ENTRY.