The Hash Table ADT

Paul English

April 30, 2013

Description

A hash table provides a collection that utilizes a hash function to enable O(1) access time to items in the collection.

Properties

1. Duplicate keys are not allowed.
2. A key may be associated with only one value.
3. A value may be associated with more than one key.
4. Keys can be compared to one another for equality; similarly for values.
5. Null keys and values are not allowed.
6. Keys can be hashed into unique values.
7. Handles collisions using a repeatable strategy, if two keys produce the same hash.

Attributes

buckets: The number of buckets used to store items in.

load factor: The threshold at which a hash table will rebalance the elements in it's buckets.

Operations

HashTable()

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

put(KeyType key, ValueType value)

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

get(KeyType key)

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

remove(KeyType key)

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

containsValue(ValueType value)

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

containsKey(KeyType key)

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

values()

pre-condition: none

responsibilities: constructor - create an empty hash map

post-condition: *size* is set to 0

return: nothing

Not shown: clear(), isEmpty(), and size()

Test Plan

Table 1: Instantiation of a **Distance** object using default values for the attributes.

Operation	Purpose	Object State	Expected Result
Distance d = new Distance()	To create a distance using the default values.	**feet = 1** **inches = 1**	A new Distance object with default values for the attributes.
d.getFeet()	To verify instantiation and accessor method.		1
d.getInches()	To verify instantiation and accessor method.		1