ADT Hash Table

HashTable= {k1, k2, k3 , ... kN}
{v1,V2, v3, ... vN}

Hash Table

Each key of the hash table has an asociated value, if two or more keys have the same position the conflict is solved by linking the elements

inv : {k1 = V1, k2 = V2, k3 = V3, kN = VN}
each key has one hash value

Primitive Operations:
-createHashtable: -> HashTable
-put: Key x Value x HashTable -> HashTable
-setLatest:Element x Key x Value x HashTable
 -> HashTable
-getValue: Key x HashTable ->Value
-existKey: Key x HashTable -> Boolean

Put(K Key, V Value) : Modifier

"Add a new value to the HashTable making a vincule with Key and Value"

{ pre: HashTable initializated }
{ post: New value added making use of HashCode vinculating the key to the respective value }

SetLatestNode(Node, K key, V value): Modifier

"Put a new LastestNode to the HashTable nodes"

{ pre: HashTable initializated }
{ post: New Node added or setted to the end of vinculated nodes in the Hash Table }

GetValue(K key):Analayzer

"Method to return the element associated to a respective Key"

{ pre: HashTable initializated }
{ post: Element associated to the respective Key in parameters or null if the element don't exists }

ExistKey(K key): Analyzer

"Method to check if a Key exists inside the HashTable"

{ pre: HashTable initializated }
{ post: True if the Key exists or False if it isn't }

CreateHashTable( ) : Constructor

"Creates a new Hash Table list where values asociated to hash codes can be added depending on the length of the HashTable "

{ pre: TRUE }
{ post: New HashTable: The new created HashTable ready to add new keys }