

HashTable

This implementation will use two hashtable instances, one for identifiers and one for constants

domain

- self.list - list of n buckets, represented as lists, n given as parameter

methods:

init(size=10):

Description: Constructor method for the HashTable class which takes the size as an optional parameter

Input: size - integer - represents the number of buckets

Output: None

add(element):

Description: Method that adds an element to the table by hashing it and adding it to the corresponding bucket

Input: The element to be added

Output: A tuple (x, y) consisting of the number of the bucket and the position in the bucket where the element was added

remove(element):

Description: Method that removes an element from the hashtable if present

Input: The element to be removed

Output: True if the element was removed / False if the element was not found

hash(element):

Description: A method that hashes an element using the unicode value for each character in a string representation of an element (works on numbers too, eg. 12 turns into '12') multiplied by its index in the string

Input: The element to be hashed

Output: The hashed value

get_pos(element):

Description: Method that returns a tuple (x, y) representing the position of the element - x being the number of the bucket and y being the position in that bucket

Input: The element

Output: The tuple

get_item(bucket, index)

Description: Method that returns the element at the given position

Input: bucket, index - bucket number and the index in that bucket

Output: Element at that position