

Link to Github repo: <https://github.com/915-Muscalagiu-AncaIoana/lab2-lftc>

Documentation

Class: Pair

- class which groups two generic objects, a key and a value

Class: SymbolsTable

- class which is responsible for keeping the tokens found by the scanner. It implements the hash table data structure with separate chaining collision handling technique, the hash table is represented as a list of list, the larger list has a maximum fixed size (which is used also for the hashing) and an actual size, the number of elements added until a certain point

- function which hashes an element based on the number of slots available in the hash table and returns the position the element in hashed on within the hash table

element : the element to be hashed

Integer hash(T element);

- function which adds an element to the hash table if it is not already within the hash table. The function checks the existence of the element in the table, if it already exists then the position of the element is returned, otherwise the element is added on the position it is hashed on in the larger list, at the end of the list on that specific position.

element : the element to be added to the hash table

Pair<Integer,Integer> add(T element)

- function which searches for an element in the hash table. The function computes the position of the element in the hash table using the hash function. If the element given is found in the list on that position then its position in the hash table is returned, otherwise the function returns null.

element : the element to be looked up in the hash table

Pair<Integer,Integer> lookup(T element)

- function which returns the number of elements added to the hash table until now

Integer getSize()