Laboratory 3

Symbol Table

Mihai Circu

Group 932

GitHub link:
https://github.com/CMihai998/FLCD/tree/master/Symbol%20Table

I implemented the symbol table using hash table and for collisions I am hashing to lists (since I have used python), the lists are going to be generated as they grow.

Hash Table<T>:

int index (T elem)

input: elem whose index to find

returns: - index where elem hashes in table if elem is in hash table

- -1, if elem is not in hash table

complexity: O(1)

int add (T elem)

input: elem to be added

returns: - index of elem, if elem already in hash table

- index of elem where it was added, if not already in hash table

complexity: O(1)

int hash (T elem)

input: elem whose hash to compute

returns: - hashed value of elem

complexity: O(1)

Symbol Table:

int insert (T elem)

input: elem to be added

returns: - index where elem was added

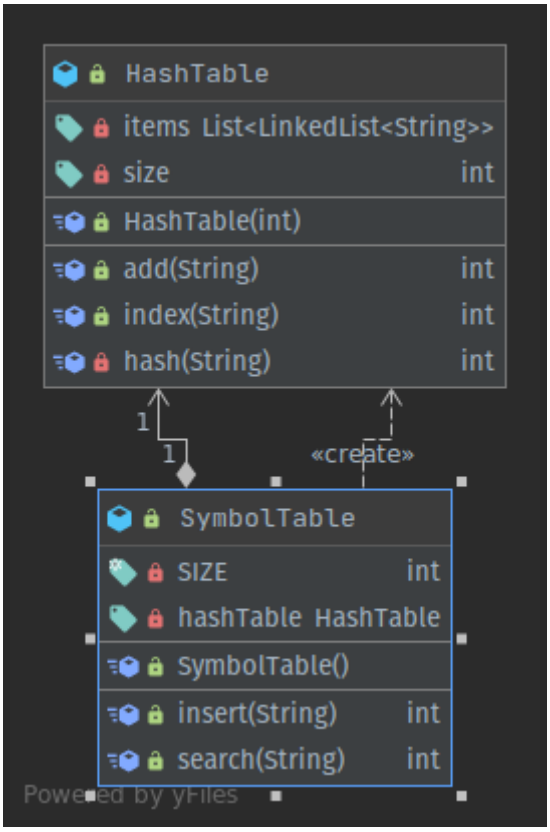
- index where elem is already situated

int index (T elem)

input: elem whose index to find

returns: - index where elem hashes in table if elem is in symbol table

- -1, if elem is not in symbol table


HashTable
items List<LinkedList<String>>
size int
HashTable(int)
add(String) int
index(String) int
hash(String) int
1
1
«create»
SymbolTable
SIZE int
hashTable HashTable
SymbolTable()
insert(String) int
search(String) int
Powered by yFiles