https://github.com/911-Ciurcau-Leonardo-Iulian/FLCD/tree/Lab2SymbolTable

```
+class SymbolTable
    -hashTable: HashTable<String, Integer> # the hash table used for storing the symbols
    +static globalPosition: Integer # used for storing the positions of symbols

    +SymbolTable()
    +add(symbol: String) # adds a symbol to the symbol table with the current globalPosition as value, then increments globalPosition
    +contains(symbol: String): boolean # return true if a symbol exists in the table, false otherwise
    +getPosition(symbol: String): Integer # gets the position of the symbol if it exists, -1 otherwise

+class HashTable<K, V>
    -struct HashBucket
        key: K
        value: V
        next: HashBucket* # next in the bucket chain

    -static constant loadFactorLimit: double = 0.7
    -size: Integer # the total number of buckets
    -count: Integer # the total number of current elements
    -buckets: HashBucket**

    -hash(value: String): Integer # computes the hash of a string value, the sum of ascii codes modulo size
    -loadFactor(): double # computes the current load factor as count / size
    -loadFactorExceeded: boolean # returns true when loadFactor() > loadFactorLimit, false otherwise
    -resize() # resizes the hash table for when the load factor is exceeded by doubling its size and reinserting the elements
    +HashTable()
    +add(key: K, value: V) # adds the key and the value in the hash table
    +contains(key: K): boolean # returns true if key is found in the hash table, false otherwise
    +get(key: K): V* # returns a pointer to the value mapped to the key if it exists, null otherwise
    +~HashTable()
```