Github link: https://github.com/PopFelix/LFTC

HashTable:

Size(int) = defines a fixed size for the hash table

Buckets Array<Array>> = a list of lists, each bucket having assigned a list. The declarations are performed in the constructor

hashFunction(element): int = returns an absolute of the element's hashCode, computed by the java function inherited from the Object main class modulo the bucket's size. It is used each time we add or search an element in the HashTable, in order to see in which bucket it will be found.

getByPos(bucketPosition:int , listPosition:int): element = this function will first find the bucket associated with the first parameter and then it will return the element at index listPosition from that bucket. It is used when printing the whole HashTable, in order to get the element from those indexes and print it.

containsElement(element: T): Boolean = true if we find the specified element in the hashTable, otherwise false

add(element: T): void = the program will first check if the element already exists in the HashTable, in which case it will not do any operation. If the element was not found, it will add it to the table.

SymbolTable: a class which contains two hash tables as private attributes. It is composed of an identifier table and table for constants.

Size(int) = defines a fixed size for the two hashTables

addIdentifier(identifier: String) = will add the string of the identifier in the respective table

addConstant (constant: String) = will add the string of the constant in the respective table