https://github.com/915MunteanSimon/FLCD/tree/main/lab_3/lab_2

For my SymbolTable I chose to implement one HashTable which can be used for both the identifiers table and constants table, as well as one single table which contains both of them.
The approach for my hashtable is o have lists in lists, so basically, on each position from the list I have another list.

getSize() method returns the size of the hashtable

findPositionOfTerm(String term) method returns the position of the term, which is computed the following way: we compute the hash for that term, we check whether the list from that is not empty, if it is empty we return null, otherwise we take the list from that position, parse it and when we reach our terminal, we create a Pair with the hash and terminal's index from the list.

findByPos(Pair pos) method will return the terminal from the symbol table based on its position

hash(String key) – represents my hash function and the approach used is to compute the sum of the characters and do modulo size.

containsTerm(String term) – returns true or false, depending if the element belongs to the symbol table or not

add(String term) – adds an element to the symbol table

All those methods, expect the hash method are implemented in the SymbolTable class.
I have also defined the Pair as a data structure. It is composed of 2 integer numbers.