https://github.com/eduard240300/Lab-FLCD

Symbol Table

> Uses a list as an hashtable and the variables: "positions" - the number of positions available in the hashtable and "occupied_positions" - the number of positions currently occupied in the hashtable When a new element is added it's hashcode is calculated, if the position is empty, the element is added. If the position is not empty a new hashcode is calculated until a position is empty. When the hashtable reaches 75% occupied positions the capacity is increased

Interface:

- `constructor`
- `resizeHashTable() -> ()` Resizes the hashtable increasing the capacity by a factor of 2.
- `hash(symbol : string, index : int) -> int` Returns the hashcode asociated to a symbol using the formula h(symbol)=(h'(symbol)%capacity+index)%capacity
- `find(symbol : string) -> int` Returns the position of the symbol searched or Null if the symbol is not in the hashtable
- `add(symbol : string) -> int` Adds the position on which the element was added, or in case the element was already in the hashtable the position on which the element is.