# Symbol Table Implementation

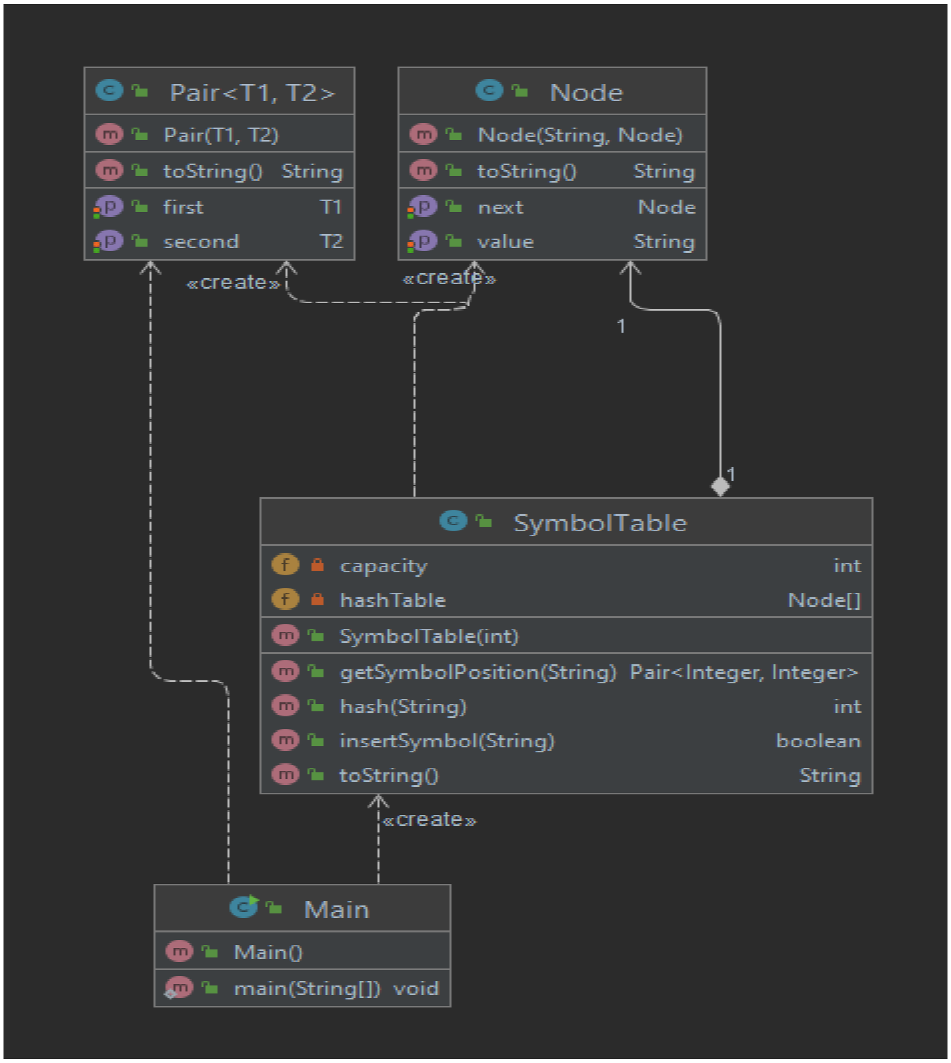
## Github link

https://github.com/CimpeanAndreea/FLCD/tree/master/Lab2

## Representation

The structure of the Symbol Table is a Hash Table with separate chaining as the method to solve the collisions.

To represent and to test this approach the project is structured as shown in the following class diagram:



Analyzing the class diagram, 4 classes can be noticed:

1. Main

   This contains only the main function in which some tests are run in order to ensure the correctness of the implementation.

2. Node

This class is used to represent a node of a linked list. It has the following properties:

value: string (an actual symbol in the symbol table)<br>
next: Node (link to the following Node in the list)

3. Pair

This is a helper class to represent the position in the symbol table of a symbol

4. SymbolTable

This class represents the symbol table with its corresponding operations, being a customized hash table.

It has the following properties:

hashTable: (Node)[] (an array containing the first node of each of the linked lists)

capacicity: integer (the size of the hashTable array)

It implements the following functions:

function hash(symbol)

description: from a symbol to be inserted in the table obtain a position of a linked list in the hashTable array to add it there, this is achieved using the sum of the ASCII codes of characters modulo the capacity of the hash table

pre: symbol is a string

post: return the hash value of the symbol as an integer

function insertSymbol(symbol)

description: insert a symbol in the symbol table if it is not already there

pre: symbol is a string

post: symbol is in the table if it was not previously added and the function returns true or the function returns false if it was previously added

function getSymbolPosition(symbol)

description: if symbol exists in the symbol table then get the position of a it as a pair (first, second), where first is the position of the containing linked list in the hashTable array and second is the position of the containing Node inside the linked list

pre: symbol is a string

post: returns a Pair with the position if the symbol exists in the table, otherwise returns null