GitHub link: https://github.com/hognogicristina/FLCT/tree/main/Lab2

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.

HashTable

The HashTable class represents a basic hash table data structure. It is initialized with a specified capacity, which determines the number of buckets in the hash table. Buckets are used to store key-value pairs. This function creates an empty hash table with the specified capacity.

get_capacity(self) - This method allows you to retrieve the capacity of the hash table. It returns the number of buckets in the hash table.

hash(self, key) - The hash method is responsible for calculating the hash value of a given key. If the key is an integer, it calculates the hash using the modulo operation. If the key is a string, it employs a hashing algorithm that iterates through the characters of the string to compute a hash value.

contains(self, key) - The contains method checks if a given key exists in the hash table. It calculates the hash value for the key and then searches for the key within the corresponding bucket. It returns True if the key is found, and False if it is not found.

get_hash_value(self, key) - This function retrieves the hash value for a given key. It distinguishes between integer and string keys and calculates the hash value accordingly.

add(self, key) - The add method adds a key to the hash table. It calculates the hash value for the key and appends it to the corresponding bucket if the key is not already present in the table.

get_position(self, key) - This method returns the position (bucket index) where a given key would be stored in the hash table.

SymbolTable Class

The SymbolTable class is a specialized version of the HashTable class that is used as a symbol table. It inherits all the functions from the HashTable class and adds some specific methods for managing symbols and their attributes.

add_symbol(self, symbol, attributes) - is used to incorporate a symbol, along with its associated attributes, into the symbol table. It computes the hash value for the symbol, and then proceeds to insert the symbol and attributes at the outset of the relevant bucket, thereby facilitating straightforward retrieval of attributes for a given symbol. This enables the symbol table to store symbol-attribute pairs for later access and manipulation, offering an efficient and structured approach for managing symbol data with their respective attributes.

get_attributes(self, symbol) - This function retrieves the attributes of a specified symbol from the symbol table. It calculates the hash value for the symbol and searches for the symbol within the corresponding bucket. If found, it returns the attributes associated with the symbol; otherwise, it returns None.