Documentation

https://github.com/927TriponBriana/University-Projects/tree/main/Semester%205/FLCD

For my SymbolTable I chose to implement 2 separate HashTables, one for identifiers and one for constants. My HashTable is generic, representing the data type of the object that will be going to be inserted into the table. The variable “table” is an array of lists, that serves as the underlying data structure to store values. Each element of the array is a “bucket” that can hold multiple values.

hashFunction(Value value) – I computed the sum of the ASCII codes of the characters and made modulo the size of the HashTable;

add(Value value) – adds an element to the SymbolTable if the position computed by the hashFunction() is not occupied, otherwise it will compute a new position until it finds one that is not occupied;

contains(Value value) – checks if a given element exists in the SymbolTable;

getHashTable() – returns a list of the values from the SymbolTable;

getPosition(Value value) – return the position of a specific element from the SymbolTable;

getBucketPosition(Value value) – return the index of the bucket where the given element would be stored;