Documentation

The Symbol Table is composed from 3 separate hash tables, one for identifiers, one for integer constants and one for string constants. Each hash table is represented by a list in which every position is another list, in order to be able to store values that hash to the same position. Also, the hash tables have a size. An element from the symbol table has as position a pair of indices, the first one being the index of the list in which the element is stored and the second one being its actual position in that list. The hash function is value modulo the size of the list, for integer values, and the sum of the ASCII codes of the characters modulo the size of the list, for string constants/identifiers. The implementation of the hash table is generic.

Operations:

Hash table

- hash(key: int): int – computes the position in the ST of the list in which the integer constant will be added
- hash(key: string): int – computes the position in the ST of the list in which the string constant/identifier will be added, based on the sum of the ASCII codes of their characters
- getSize(): int – return the size of the hash table
- getHashValue(key: T): int – return the corresponding position in the ST according to the type of the parameter 'key'
- add(key: T): (int, int) – add the key to the hash table and return its position if the operation is successful; otherwise, throw an exception
- contains(key: T): boolean – return if the given key is in the hash table or not
- getPosition(key: T): (int, int) – return the position in the ST of the given key, if it exists; otherwise, return (-1, -1)
- toString() (overridden method) – return the string representation of the hash table

Symbol table

- has 3 hash tables: for identifiers, for string constants and for integer constants
- addIdentifier(name: string): (int, int) – add an identifier and return its position in the ST
- addIntConstant(constant: int): (int, int) – add an integer constant and return its position in the ST
- addStringConstant(constant: string): (int, int) – add an string constant and return its position in the ST
- hasIdentifier(name: string): boolean – return if the given identifier is in the ST or not

- hasIntConstant(constant: int): boolean – return if the given integer constant is in the ST or not
- hasStringConstant(constant: string): boolean – return if the given string constant is in the ST or not
- getPositionIdentifier(name: string): (int, int) – get the position of the identifier in the ST
- getPositionIntConstant(constant: int): (int, int) – get the position of the integer constant in the ST
- getPositionStringConstant(constant: string): (int, int) – get the position of the string constant in the ST
- toString() (overridden method) – get the string representation of the whole symbol table