

Github repo:<https://github.com/cs-ubbcluj-ro/lab-work-computer-science-2024-iuliamariagroza/tree/main/2-Finite-Automata>

### Documentation:

The Symbol Table is composed of 2 separate hash tables, one for identifiers and one for constants. The hash table uses separate chaining method to solve collision problems which is done by storing multiple elements of the same hash value in a list. Each element from the symbol table has as its position a pair of 2 values: the hash value (the index) and the position within the list at that index. There are 2 hash functions for each of the types: integers and strings. The integer hash function is the value of the element modulo the size of the list while the hash function for the strings is calculated by summing the ASCII values of the characters in the string and then taking the result modulo the size of the list.

### Operations for each class:

#### 1. Hash table

a. `int generateHashForIntegers(int key)`

—> generates the hash value for the integers

b. `int generateHashForStrings(string key)`

—> generates the hash value for the strings

c. `boolean containsElement(T key)`

—> returns true if a given key exists in the hash table, false otherwise

d. `int computeHashValue(T key)`

—> determines which has function to use depending on the type of key

e.add(T key)

—> inserts a key into the hash table and returns a pair containing the hash value and the position within the list of that hash value

f. getPosition(T key)

—> returns the position of a key if found, -1 if not

g. toString()

—> provides a readable representation of the hash table

## 2. Pair

a. getFirstElement()

—> returns the first element of the pair

b. getSecondElement()

—> returns the second element of the pair

c. toString()

—> provides a readable representation of the pairs

## 3. Symbol table

a. addIdentifier(string identifier)

—> adds an identifier to the symbol table

b. addConstant(string constant)

—> adds a constant to the symbol table

c. boolean containsIdentifier()

—> checks if an identifier exists in the symbol table

d. boolean containsConstant()

—> checks if a constant exists in the  
symbol table

e. findIdentifierPosition(string identifier)

—> returns the position of the identifier in  
the symbol table

f. findConstantPosition(string identifier)

—> returns the position of the constant in  
the symbol table

g. toString()

—> provides a readable representation  
of the symbol table