Documentation ~ Symbol Table // Hashtable

Formal Languages and Compiler Design

Bican Iulia, 931/2 - Lab2

Link to Github Repository:

https://github.com/iuliabican/Formal-Languages-and-Compiler-Design

Problem Requirements:

Implement the Symbol Table (ST) as the specified data structure, with the corresponding operations.

Symbol Table

The underlying structure of the Symbol Table is a custom Hash Table, due to the fact that its operations run in O(1). The SymbolTable class will be considered, for the moment, a CustomHashTable.

The hash function is computed as follows:

- Sum = 0;
- For each char of the string, its ascii code is added to the sum;
- Sum % SIZE is returned.

The operations implemented by this CustomHashTable are:

1. *insert (identifier)* - Adds the given identifier into de symbol table

Collision problems are solved using Linear Probing.

- Params: String identifier
- Returns: TRUE if the identifier doesn't exist in symbol table

FALSE otherwise

2. *find (identifier)* - Searches for the identifier in the symbol table

- Params: String identifier
- Returns: position (INTEGER) if the identifier exists in symbol table

-1 otherwise