Github repo: https://github.com/915-Petruta-Razvan/LFTC_Labs

Lab2_LFTC

Documentation

The Symbol Table is composed of 3 separate hash tables:
- for identifiers
- for int constants
- for string constants

Hash Table

Details: the implementation of this data structure is generic. A hash table has a predefined size (capacity) and is built as a list of lists, such that an element in the list has 2 indices, first is the index of the list in which the element is stored and the second index represents the index in that list.
In order to determine the first index, we use a hash function for the elements depending on the type of the element:
- for integers the hash is the size % element
- for strings (that includes identifiers as well) the hash is (the sum of ASCII code of every character) % size

After we determine the first index, we push the element in the list corresponding to that list

Methods:
- Hash(key: int): computes the hash index for an int element (as specified above)
- Hash(key: string): computes the hash index for an string element (as specified above)
- Add(key: T): adds an element to the hash table and returns its position if the operation is successful, otherwise it throws an exception
- Contains(key: T): checks if the element is present in the hash table and returns true, otherwise it returns false
- GetPosition(key: T): checks if the element is present in the hash table and returns its position, otherwise it returns (-1, -1)
- ToString(): overrides the ToString method and adapts it to a nicer format

Symbol Table

Details: contains 3 hash tables that will be initialised with a predefined capacity: for identifiers, int constants and string constants.

Methods:

- AddIdentifier(name: string): adds an identifier to the identifiers hash table and returns its position or throws an exception if the operation could not be completed
- AddIntConstant(constant: int): adds an int constant to the int constants hash table and returns its position or throws an exception if the operation could not be completed
- AddStringConstant(constant: string): adds an string constant to the string constants hash table and returns its position or throws an exception if the operation could not be completed
- HasIdentifier(name: string): checks if an identifier is present in the identifiers hash table
- HasIntConstant(constant: int): checks if an int constant is present in the int constants hash table
- HasStringConstant(constant: string): checks if a string constant is present in the string constants hash table
- GetIdentifierPosition(name: string): returns the position of an identifier in the identifiers hash table
- GetIntConstantPosition(constant: int): returns the position of an int constant in the int constants hash table
- GetStringConstantPosition(constant: string): returns the position of a string constant in the string constants hash table
- ToString(): overrides the ToString function and adapts it to a nicer format, in which we can see each hash table