LAB 2

SymbolTable is a class that allows you to save tokens (string values). Each token is given an unique index. When you add a token to the SymbolTable, if the token already exists, it returns its position, otherwise a new unique position is assigned to it. Possitions are assigned in increasing order.

The SymbolTable uses a custom hash table named HashTable to store its data. HashTable is implemented using a bucket array of 1M elements. Each bucket is a System.Collections.Generic.List<KeyValuePair<TKey, TValue>>.

GitHub: https://github.com/Alin-St/FCLD (https://github.com/Alin-St/FCLD)

LAB 3

The class that performs scanning the file is Scanner. The Scanner has a SymbolTable for storing both identifiers and constants, and a ProgramInternalForm instance for storing all types of tokens. The Scanner has a method named Scan that takes as input the content of a file and updates its SymbolTable and PIF based on the content of the file.

The PIF is a list of (token, position) pairs. The token can be an operator, separator, keyword, identifier or constant. The position is the position in the SymbolTable of the token if it is an identifier or constant, or -1 otherwise.

In order to perform the lexical analysis, the Scanner tries to continuously match the beginning od the file with a regular expression for each type of token possible. If a match is found, the token is added to the PIF, and the file is updated by removing the token.

The Regexes used for matching identifiers and constants are (found in Scanner.cs):

- identifier regex: `"^[a-zA-Z][a-zA-Z0-9_]*"`
- integer regex: `"^[0-9]+"`
- boolean regex: `"^(true|false)"`