https://github.com/Pop-Vlad/FLCD

Symbol table:

The symbol table uses a hash table. Each position of the table is a reference to an element of type Node. Collisions resolution: Store tokens that hash to the same position in a double linked list of nodes.

Symbol table structure:

- Table: Node[] – The table in which the list of nodes are stored. Each node contains a token.
- Size: int – The size of the hash table

Node structure:

- Next: Node – Reference to the next node in the linked list
- Previous: Node – Reference to the previous node int the linked list
- Token: String – The token

Dispersion: The hash table uses the default java hash() function for the tokens.

Program internal form:

The PIF uses a list of pairs. Each pair has a string representing the token and an integer representing the position. If the token is an identifier “0” is used for the token. If it is a constant “1” is used for the token. If the token is not an identifier or constant -1 is used for the position.

PIF structure:

- Pairs: Pair[]

Pair:

- First: string
- Second: int

Scanner:

The source code is stored in a string called “program” which is analyzed character by character.

The function detect() detects 1 token:

- The scanner starts from a position and builds a token by adding 1 character at a time.
- The scanner looks ahead if the token it builds can become a valid token. This is done by checking if an element from token.in starts with the current string or if the current string is a valid identifier or constant.

- If the token is “+” and “-“, the scanner checks if it is an operator(e.g. 1 + 2) or is part of an integer constant (e.g. x := +3). This is done by checking if the last token in the PIF is “:=”. If so, the token is part of a constant.

After detect() finds a token, the scanner tries to classify it as:

- Reserved word, operator or separator – The scanner searches for the token in a list of strings given by the file token.in
- Identifier or constant: - The scanner checks if the token matches a given regex.

The classified value is added to the PIF.

Class diagram: