Github: https://github.com/Szabo-Mark/FLCD

The **symbol table** is just a wrapper over the HashTable:

HashTable implementation:

The class **Position** will hold the position of the bucket array and the position in the linked listed of the token.

The class **Node** is a node in the linked list. It holds the value(the actual token) and the next node in the list.

The class **HashTable** has the public method add(String token).

It works in the following way:

- if the given token is already in the symbol table it will return the position of it(a Position object)
- if not, it firstly adds the token into the table and then returns its new position.

The collision strategy is linked lists. Each slot in the table is the head of a linked list.

First we hash the given token and get the head of the linked list on that position.

We parse that linked list searching for the token. If it's not there, we add it at the end.

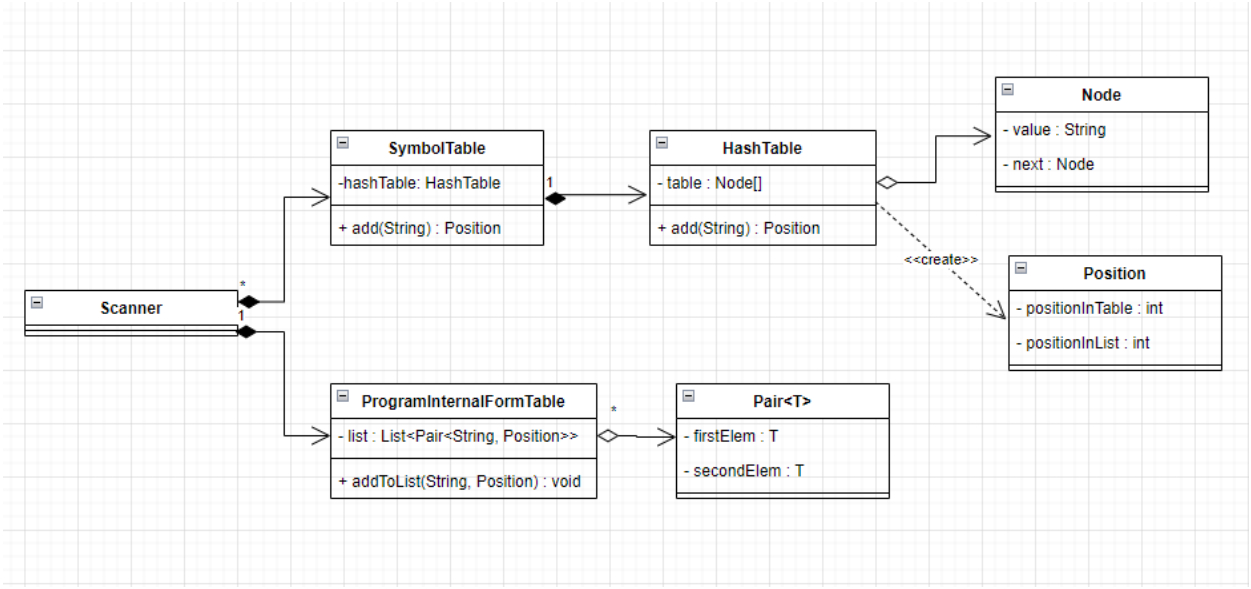
The **program internal form** is a wrapper over a list which holds pairs of String and Position.

- The String is either the token(in case it's in the token.in file) or "const"/"id" depending on the case.
- The Position is the position returned by the SymbolTable when adding the token.
-

RegEx used for token matching:

- Separators – " |;|\{|}|\[|]|!|<|>|=|-|\+|*|/|%|\t"
  Separators regex – "((?=" + SEPARATORS + ")|(?<=" + SEPARATORS + "))"
  - Tokenizes the string based on the given separators. Using look-ahead and look-behind it keeps the separators as tokens.
- Integers - "^0|[+|-]?[1-9][0-9]*$"
  - Matches all integers. Signed or unsigned.
- Characters - "^'[a-zA-Z0-9]'$"
  - Matches all character constants. A character can be a lowercase or uppercase letter or a digit. Both between single quotes.

Diagram


Node
- value : String
- next : Node
SymbolTable
-hashTable: HashTable
+ add(String) : Position
1
HashTable
- table : Node[]
+ add(String) : Position
<<create>>
Position
- positionInTable : int
- positionInList : int
Scanner
*
1
ProgramInternalFormTable
- list : List<Pair<String, Position>>
+ addToList(String, Position) : void
*
Pair<T>
- firstElem : T
- secondElem : T