Custom Scanner documentation

Link to github: <https://github.com/IuliaPapureanu/FLCD>

Diagram

Description automatically generated

**Class HashtablePosition:**

-hashPosition (int): the hashcode of a certain key

-slot(int): the position of a certain key in a list with the same hashcodes

**Class SymbolTable :**

-implemented using hashtable

* size (int) : the size of the hastable / nr of “buckets” the table uses – works best with prime number
* elements (ArrayList (ArrayList)): here we store each element

-private int hash(String key):

-sums the ascii code of each char in the given key and divides it by the size of the hashtable

-returns the remainder of the division

-public HashtablePosition add(String key):

-checks if the key already exists in the symbol table

- if it doesn’t, the key is added

-returns the position of the given key in the table

**Class PIF**

-(Program internal form) an arrayList of PIFelements( a PIFelements consists of a String – the id of the element- and a position in the symbol table – a pair of 2 integer)

-public void add() – adds and element in the pif list

**Language**

-here we define the operators, keywords and separators. We also define the pattern for identifiers, numerical values and string values.

|  |  |
| --- | --- |
| -public Boolean IsOperator()  -public Boolean IsKeyword()  -public Boolean IsSeparator()  -public Boolean IsIdentifier()  -public Boolean IsOperator()  -public Boolean IsConstant() | All these functions check if the given string matches their respective pattern(identifier, numerical value, string value), or is part of the  according list (operator,keyword,separator) |

**Class CustomScanner**

* reads given file, constructs PIF and SymbolTable

- ls : Language

-pif :PIF

-capacity : int – size of the symbol table

-symbolTable : SymbolTable

* public void scan() – reads the contents of a given file and breaks each line down into tokens, building the symboltable and the pif accordingly
* public ArrayList<String> tokenize(String line) – breaks down the line and identifies the operators and constants
* public void buildPIF() – constructs the pif with the list of tokens given
* public void writeResults() – writes the PIF and SymbolTable of the program in 2 output files