git: https://github.com/915-Negrila-Iulia/FLCD/tree/main/lab2/src

Symbol Table

- defining ST:

- stores information about symbolic names and their associated positions from the source of the program, during compilation time

- ST organization:

- the symbol table provided is organised as a hash table with a given size

- hash function used receives the symbol(as a string) and based on its characters ASCII sum returns a hashKey

- in order to avoid collisions, the table consists of an array having multiple arrays, so even if multiple symbols have the same hashKey in the outer array, they will be stored one after another in the inner array

- therefore, position of a symbol is represented by a tuple containing: the hashKey and the index from the inner list that is reffered by the hashKey

- ST functionalities:

- a new ST object can be created by providing a size for the outer array (preferably a prime number)

- int hash(String key): hash function described previously

- boolean isKey(String key): checks if a given symbol already exists in ST; returns true/false

- AbstractMap.SimpleEntry<Integer, Integer> addSymbol(String symbol): adds a new symbol in ST; returns its position(mentioned previously) or an invalid position if an error occures

- it should be mentioned that a symbol that has a new hashKey(does not exists in ST already) will be added on position 0 of the inner array list(located at hashKey position), otherwise the symbol will pe added after the last element of that inner list

- String toString() function is overriden and displays each inner array with associated symbols on a separate line

PIF

- defining PIF:

- stores information about tokens and their position in ST

- PIF organization:

- organised as an arrayList containing pairs; A pair consits of a pair representing position in ST hashTable and a String representing the definition of a symbol (id,const, or the token from token.in)

- PIF functionalities:

- add(AbstractMap.SimpleEntry position, String symbol) adds the pair in PIF

- toString() function is overriden and displays the pairs in PIF line by line; returns string

Scanner

- defining Scanner:

- splits each line from the program by given delimiters and checks for lexical errors by using regexes

- Scanner organization:

- regex used:

identifierRegex = "^([a-zA-Z][a-zA-Z\\d]\*)$" checks if the symbol is a valid identifier; any sequence of letters and numbers that starts with a letter is valid

digitRegex = "-?[1-9]\\d\*|0" checks if a symbol is a valid number; any number that does not start with 0 or 0 itself is a valid number

charRegex = "'[a-zA-z\\d]'" checks if a symbol is a valid character; any single character or single digit between '' is a valid char

delimitersRegex = "\\s|;|," delimitars used for splitting lines; space ';' or ',' is a delimiter

- definition of a symbol -> used for identifiers and constants, is used in PIF

String ID = "id"

String CONST = "const"

- Class objects used:

pifTable of type PIF

identifTable of type SymbolTable used for identifiers

constSymbolTable of type SymbolTable used for constants

- program File to get the program that need to be scanned is

stored in String programFile variable

- Scanner functionalities:

- scanner is created with given SymbolTables for identifiers and constants, PIF and path to file where program is stored

- isIdentifier(String symbol) checks if given symbol is a valid identifier by using identifierRegex and returns a boolean value

- isConstant(String symbol) checks if given symbol is a valid constant by using constantRegex and returns a boolean value

- isToken(String symbol) checks if given symbol is a valid token from file "token.in" and returns a boolean

- addST(String symbol) adds the given symbol to the identifTable if it's an identifier or to the constSymbolTable if it's a constant

- addPIF(String symbol) adds pair in PIF such that:

for token the position is (-1,-1) and the string is the token itself

for identifier the position from identifTable ST and the string "id"

for constant the position form constSymbolTable ST and the string "const"

if it is not a valid identifier/constant/token throws an excpeption: "Lexical error" for the symbol

- scanLine(String line) scans the given line and checks the symbols, and add it to ST and PIF

- scan() scans the program from file line by line and throws lexical error mentioning the number of the line if needed

- outputScanner(String filename) writes the tables of the scanned program to the given file

- outputTable(String tableStr, String filename) writes given table after scan to given file

- toString() function is overriden and displays the ST and PIF tables for a program after a scan; returns string

