ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΘΝΙΚΌ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΌ ΠΑΝΕΠΙΣΤΉΜΙΟ ΑΘΗΝΏΝ
ΤΜΉΜΑ ΠΛΗΡΟΦΟΡΙΚΉΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΏΝ

Τίτλος Εργασίας

1η Προγραμματιστική Εργασία

Implementation of a LL(1) parser for a simple calculator and a translator to Java for a simple language

Μάθημα

Compilers, K31 (Μεταγλωττιστές, K31)

Εαρινό εξάμηνο 2021-22

Ονοματεπώνυμο φοιτητη:

- Μαυραπίδης Νικόλαος (Α.Μ.: 11152017 00082)

Αθήνα, 2020

Part 1

Η **γραμματική** που χρησιμοποιήθηκε για τον χειρισμό των εκφράσεων της γλώσσας στο part 1.

expr ::= term expr2
expr2 ::= + term expr2
| - term expr2
| ε
term ::= factor term2
term2 ::= ** factor term2
| ε
factor ::= num
| (expr)
num ::= 0
| digit_without_0 num2
num2 ::= digit num2
| ε
digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
digit_without_0 ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

ΕΚΤΈΛΕΣΗ ΚΑΙ ΧΡΉΣΗ ΠΡΟΓΡΆΜΜΑΤΟΣ

Ο παρεχόμενος φάκελος συμπεριλαμβάνει τα αρχεία **CalcEvaluator**, **Main** και **ParseError**. Με τις εντολές **javac Main.java** και **java Main** το πρόγραμμα μεταγλωττίζεται, εκτελείται και **περιμένει είσοδο** από τον χρήστη. Στο τέλος της έκφρασης που δίνεται από τον χρήστη η είσοδος μπορεί να τερματιστεί είτε με **n** είτε με **EOF**.

Part 2

Για αρχή θα παραθέσω την **γραμματική** που χρησιμοποιήθηκε για τον χειρισμό των εκφράσεων της γλώσσας στο part 2.

program ::= declarations main
declarations ::= declaration declarations
| ε
declaration ::= DECLARATION_FUNCTION_START expression }
main ::= expression main
| ε
expression ::= if_statement
| concatenation
| function_call
| IDENTIFIER
|STRING_LITERAL
if_statement ::= IF condition_function) expression ELSE expression
condition_function ::= expression condition_function_tail
condition_function_tail ::= SUFFIX expression
| PREFIX expression
concatenation ::= expression PLUS expression
function_call ::= FUNCTION_START args)
args ::= expression args_tail
| ε
args_tail ::= COMMA expression args_tail
| ε

Στην παραπάνω γραμματική χρησιμοποιήθηκαν δύο **τερματικά σύμβολα - tokens** DECLARATION_FUNCTION_START και FUNCTION_START. Ο κύριος λόγος ύπαρξης τους είναι για τον διαχωρισμό της απλής κλήσης συνάρτησης και της δήλωσης – ορισμό συνάρτησης. Ωστόσο **και οι δύο** κανόνες ξεκινάνε με ένα IDENTIFIER (το όνομα της συνάρτησης) και ένα άνοιγμα παρένθεσης(. Η βασική **ειδοποιός διαφορά** αυτών των δύο στη συγκεκριμένη εργασία είναι η ύπαρξη αγκύλης {. Συνεπώς, δημιουργήθηκαν από τον **Lexer** δύο διαφορετικά σύμβολα μέσω κανονικών εκφράσεων για τον διαχωρισμό τους και για την λύση των **ambiguity προβλημάτων** που προέκυπταν. Τα υπόλοιπα τερματικά σύμβολα – tokens IDENTIFIER, STRING_LITERAL, SUFFIX, PREFIX, PLUS, COMMA έχουν ξεκάθαρη λειτουργία.

Επίσης στο πρόγραμμα ορίζεται και η προτεραιότητα αυτών των tokens ως εξής:
precedence(if) < precedence(prefix/suffix) < precedence(concat) < precedence(function_call).

ΕΚΤΈΛΕΣΗ ΚΑΙ ΧΡΉΣΗ ΠΡΟΓΡΆΜΜΑΤΟΣ

Ο παρεχόμενος φάκελος συμπεριλαμβάνει ένα **makefile** ίδιο με αυτό που μας δόθηκε από το υλικό του εργαστηρίου για την μεταγλώττιση και την εκτέλεση όλων των αρχείων. Επίσης παρέχεται το αρχείο **input.txt** απ'όπου το πρόγραμμα διαβάζει την είσοδο. Ομοίως η έξοδος του προγράμματος δηλαδή το παραγόμενο **java** αρχείο αποθηκεύεται στον φάκελο **output/** όπου ο χρήστης μπορεί να μεταβεί και να μεταγλωττίσει και τρέξει το πρόγραμμα με τις εντολές **javac Main.java** και **java Main** αντίστοιχα.