

# Σχεδίαση Γλωσσών Προγραμματισμού (Εαρινό Εξάμηνο 2020-2021)

## Προαιρετική Εργασία

Ονοματεπώνυμο: Μπαρακλilής Ιωάννης

AEM: 3685

email: [imparakl@csd.auth.gr](mailto:imparakl@csd.auth.gr)

## Εργαλεία Ανάλυσης

Ως εργαλεία ανάλυσης χρησιμοποιήθηκαν:

- Το εργαλείο flex για την παραγωγή κώδικα λεξικής ανάλυσης. Για την περιγραφή του λεξικού αναλυτή χρησιμοποιείται το αρχείο “lexer.l”.
- Το εργαλείο yacc για την παραγωγή κώδικα συντακτικής ανάλυσης. Για την περιγραφή του συντακτικού αναλυτή χρησιμοποιείται το αρχείο “parser.y”.

## Περιγραφή αναλυτών

- Για τον λεξικό αναλυτή:  
Το αρχείο “lexer.l” περιγράφει λεξικό αναλυτή ο οποίος αναγνωρίζει τις λεκτικές μονάδες που ορίστηκαν στις προδιαγραφές της γλώσσας και επιστρέφει τα αντίστοιχα αναγνωριστικά στον συντακτικό αναλυτή. Οι αριθμητικές τιμές που αντιστοιχούν στα αναγνωριστικά ορίζονται από το αρχείο “y.tab.h” το οποίο παράγεται με την παραγωγή του συντακτικού αναλυτή από το αρχείο “parser.y” με τον yacc. Σε κάποιες περιπτώσεις αναγνωριστικών (αυτά που αντιπροσωπεύουν τιμή, π.χ. οι σταθερές ακεραίων ή το όνομα μίας μεταβλητής) αποθηκεύεται στο yyval.valueString, η συμβολοσειρά του αναγνωριστικού που εντοπίζεται.
- Για τον συντακτικό αναλυτή:  
Το αρχείο “parser.y” περιγράφει συντακτικό αναλυτή ο οποίος ελέγχει την συντακτική ορθότητα του πηγαίου προγράμματος και κατασκευάζει και τυπώνει το συντακτικό δένδρο σε αντιστοιχία με την γραμματική της γλώσσας.  
Μετά την μετατροπή της γραμματικής της γλώσσας που δίνεται στο τμήμα των συντακτικών κανόνων σε μορφή αποδεκτή από το yacc, διορθώνεται το πρόβλημα που δημιουργεί το γεγονός ότι η γλώσσα είναι διφορούμενη με τον ορισμό προτεραιότητας και προσηταιριστικότητας των τελεστών (T\_COLON, T\_OROP, T\_ANDOP, T\_NOTOP, T\_RELOP, T\_ADDOP, T\_MULOP, T\_DIVOP, T\_POWEROP, T\_LPAREN και T\_RPAREN) που συμφωνούν με τις προδιαγραφές της γλώσσας.  
Τέλος, ορίστηκε κατάλληλος κώδικας (που ενεργοποιείται κατά την συντακτική ανάλυση) και συναρτήσεις για την κατασκευή και στην συνέχεια την εμφάνιση του δένδρου στην προκαθορισμένη έξοδο.

# Παραδείγματα τμημάτων εκτέλεσης

- Τμήμα αποτελέσματος εκτέλεσης του παραδείγματος “test1.f” που δίνεται στις προδιαγραφές της γλώσσας / εκφώνηση της εργασίας:

program

|---- body

| |---- declarations

| | |---- declarations

| | | |---- declarations

| | | | |---- declarations

| | | | | |---- declarations

| | | | | | |---- declarations

| | | | | | |---- EmptyDeclaration

| | | | | | |---- type

| | | | | | |---- T\_INTEGER

| | | | | | |---- vars

| | | | | | |---- vars

| | | | | | |---- vars

| | | | | | |---- vars

| | | | | | |---- undef\_variable

| | | | | | |---- listspec

| | | | | | |---- Empty

| | | | | | |---- T\_ID = x

| | | | | | |---- T\_LPAREN

| | | | | | |---- dims

| | | | | | |---- dim

| | | | | | |---- T\_ID = g

| | | | | | |---- T\_RPAREN

| | | | | | |---- T\_COMMA

| | | | | | |---- undef\_variable

| | | | | | |---- listspec

| | | | | | |---- Empty

| | | | | | |---- T\_ID = z

- Τμήμα αποτελέσματος εκτέλεσης του παραδείγματος “test2.f” που δίνεται στις προδιαγραφές της γλώσσας / εκφώνηση της εργασίας:

program

|---- body

| |---- declarations

| | |---- declarations

| | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- declarations

| | | | |---- EmptyDeclaration

| | | | |---- T\_COMMON

| | | | |---- cblock\_list

| | | | |---- cblock

| | | | |---- T\_DIVOP

| | | | |---- T\_ID = C1

| | | | |---- T\_DIVOP

| | | | |---- id\_list

| | | | |---- id\_list

| | | | |---- id\_list

| | | | |---- id\_list

| | | | |---- T\_ID = x

| | | | |---- T\_COMMA

| | | | |---- T\_ID = z

| | | | |---- T\_COMMA

| | | | |---- T\_ID = i

| | | | |---- T\_COMMA