Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Προηγμένα Θέματα Γλωσσών Προγραμματισμού
Διδάσκων: Χρήστος Γκουμόπουλος
Εργαστηριακοί συνεργάτες: Γιώργος Χρυσολωράς, Μιχαήλ Δανούσης

4η ομαδική εργασία

Οδηγίες για την παράδοση της εργασίας

Η εργασία αυτή είναι ομαδική και μπορεί να εκπονηθεί σε ομάδες έως 3 ατόμων.

Η παράδοση των εργασιών θα πρέπει να γίνει ηλεκτρονικά μέσω του GitHub στον σύνδεσμο εργασίας που αναρτάται μέσα από το eclass του μαθήματος.

Μετά την παράδοση της εργασίας, ακολουθεί προφορική εξέταση στα θέματα της εργασίας η οποία, στην περίπτωση που συνεχίζεται η εξ αποστάσεως εκπαιδευτική διαδικασία στο Τμήμα, θα πραγματοποιηθεί μέσω τηλεσυνάντησης (Zoom ή αντίστοιχο) με υποχρεωτική παρουσία των μελών της ομάδας.

Αξιολόγηση της εργασίας

Ως προς την αξιολόγηση της εργασίας θα συνεκτιμηθούν, εκτός από την ορθότητα των ερωτημάτων καθώς και την ορθή λειτουργία του συντακτικού αναλυτή, η συνεργατικότητα μέσω του αποθετηρίου καθώς και η προφορική εξέταση. Ανεξάρτητα από τον διαμοιρασμό σε επιμέρους εργασίες τα μέλη οφείλουν να γνωρίζουν σφαιρικά τα επιμέρους στοιχεία υλοποίησης των επιμέρους εργασιών συνεπώς απαιτείται και η ανταλλαγή γνώσης στα πλαίσια της ομάδας.

Περιγραφή της εργασίας

Δίνεται η ακόλουθη γραμματική μιας γλώσσας προγραμματισμού:

```
<PROGRAM>       ::= program <DECLARATIONS> <STATEMENTS> endprogram
<DECLARATIONS>  ::= ( declare <VARLIST> enddeclare )*
<VARLIST>       ::= integer variable ( , variable )* |
                    real variable  ( , variable )*
<STATEMENTS     ::= <STATEMENT> ( ; <STATEMENT> )*
<STATEMENT>     ::= input <INPUT-TAIL> |
                    print <PRINT-TAIL> |
                    variable <ASSIGNT-TAIL>
<INPUT-TAIL>    ::= ( variable )
<PRINT-TAIL>    ::= ( <EXPRESSION> )
```

```
<ASSIGN-TAIL>    ::= := <EXPRESSION>
<EXPRESSION>     ::= < SIGN> <TERM> ( + <TERM> | - <TERM> )*
<TERM>           ::= <FACTOR> ( * <FACTOR> | / <FACTOR> )*
<FACTOR>         ::= numerical_constant |
                     variable |
                     ( <EXPRESSION> )
<SIGN>           ::= ε | + | -
```

Οι λέξεις κλειδιά, τα σύμβολα δηλαδή που ανήκουν στη γλώσσα, σημειώνονται με μπλε χαρακτήρες. Για παράδειγμα, η παρένθεση μετά τη λέξη κλειδί `print` είναι με μπλε χρώμα, άρα αποτελεί μέρος της σύνταξης της `print`, ενώ η παρένθεση στον κανόνα `EXPRESSION` δεν είναι με μπλε γράμματα, διότι αποτελεί σύμβολο ομαδοποίησης που χρησιμοποιεί η γραμματική για να δείξει την εμβέλεια του συμβόλου "`*`" που υποδηλώνει επανάληψη (μετα-σύμβολο EBNF συμβολισμού).

Η γλώσσα αυτή υποστηρίζει μόνο ακέραιους και πραγματικούς αριθμούς, ενώ οι μεταβλητές της ξεκινούν από γράμμα και στη συνέχεια μπορούν να αποτελούνται από γράμματα ή αριθμούς. Στη γραμματική, με το σύμβολο `numerical_constant` συμβολίζουμε τις ακέραιες και πραγματικές σταθερές, ενώ με `variable` τις μεταβλητές του προγράμματος.

Το παρακάτω πρόγραμμα είναι γραμμένο στη γλώσσα αυτή και ζητάει την ακτίνα ενός κύκλου και στη συνέχεια υπολογίζει και τυπώνει στην οθόνη τη διάμετρο, την περιφέρεια και το εμβαδόν του κύκλου.

```
program
       declare real r, diam, area, circ enddeclare
       input(r);
       diam:=2*r
       area:=3.14*r*r;
       circ:=3.14*d;
       print(area)
       print(circ)
endprogram
```

Ερώτημα 1. Υλοποίηση Συντακτικού Αναλυτή (40%)

Ο συντακτικός αναλυτής υλοποιείται με βάση τη γραμματική. Η γραμματική είναι μορφής LL(1). Αυτό σημαίνει ότι διαβάζει την είσοδο από αριστερά στα δεξιά, αναγνωρίζει την αριστερότερη δυνατή παραγωγή και όταν βρίσκεται σε δίλλημα ως το ποιον κανόνα να ακολουθήσει της αρκεί να κοιτάξει το αμέσως επόμενο σύμβολο στην συμβολοσειρά εισόδου (top-down ανίχνευση με πρόβλεψη). Αυτό το τελευταίο στοιχείο μας ενδιαφέρει ιδιαίτερα στην κατασκευή τους συντακτικού αναλυτή.

Για την επικοινωνία του συντακτικού αναλυτή με το λεκτικό αναλυτή μπορείτε να ορίσετε μια κλάση **Token** που θα αναπαριστά μια δομή για την αποθήκευση των λεκτικών μονάδων που αναγνωρίζονται με τρία πεδία:

- `type`: Ο τύπος της λεκτικής μονάδας. Μπορεί να οριστεί μια κλάση `enum TokenType`.

- `data` (τύπου String): Η λεκτική μονάδα.
- `line` (ακέραιος): Η γραμμή στην οποία εμφανίστηκε η λεκτική μονάδα.

Να υλοποιήσετε τον συντακτικό αναλυτή είτε με το χέρι ακολουθώντας τις προδιαγραφές της προβλέπουσας αναδρομικής κατάβασης είτε χρησιμοποιώντας το εργαλείο ANTLR.

Ερώτημα 2. Σημασιολογική ανάλυση κώδικα (40%)

Προσθέστε κώδικα στον συντακτικό αναλυτή, ώστε στο τέλος της συντακτικής ανάλυσης να εμφανίζονται στην οθόνη οι εξής πληροφορίες:

1. Ο αριθμός των μεταβλητών που έχουν δηλωθεί. Πόσες από αυτές είναι ακέραιες και πόσες πραγματικές. Μεταβλητές που έχουν δηλωθεί περισσότερες από μία φορές μετριούνται σαν μία.
2. Αν (και ποια) μεταβλητή έχει δηλωθεί περισσότερες από μία φορές και πόσες.
3. Ποιες μεταβλητές (αν υπάρχουν) έχουν χρησιμοποιηθεί στο πρόγραμμα και σε ποια γραμμή του, χωρίς όμως να έχουν δηλωθεί στο τμήμα δηλώσεων (`declare-enddeclare`)
4. Ποιες μεταβλητές (αν υπάρχουν) έχουν χρησιμοποιηθεί στο πρόγραμμα και σε ποια γραμμή του, πριν γίνει σε αυτές εκχώρηση τιμής (είτε με την εντολή `input`, είτε με τον κανόνα εκχώρησης).

Για τη διαχείριση της πληροφορίας που απαιτείται για να υλοποιηθεί η σημασιολογική ανάλυση συνίσταται να αξιοποιηθούν οι κλάσεις `Variable` και `VarList` που σας δίνονται.

Ερώτημα 3. Έλεγχος Ορθής Λειτουργίας (20%)

Να ελέγξετε την ορθή λειτουργία του μεταγλωττιστή με προγράμματα τα οποία θα συντάξετε και θα είναι κατάλληλα επιλεγμένα ώστε να πείθουν για την ορθή λειτουργία όλων των παραπάνω περιπτώσεων του ερωτήματος 2.

Για παράδειγμα αν δοθεί το ακόλουθο πρόγραμμα στην είσοδο:

```
 1.  program
 2.        declare real r, r, r, diam, area, circ, circ enddeclare
 3.        declare integer K enddeclare
 4.
 5.        input(rx22);
 6.        diam:=2*rxx;
 7.        areaX:=3.14*r*r;
 8.        circ:=3.14*d;
 9.        K := K*diam;
10.        print(area);
```

```
11.       print(circ)
12.
13.  endprogram
```

Αναμένεται η ακόλουθη έξοδος από τον μεταγλωττιστή που θα πρέπει να κατασκευαστεί:

```
Variable rx22 not declared in line 5
Variable rxx not declared in line 6
Variable rxx not set before use in line 6
Variable areaX not declared in line 7
Variable r not set before use in line 7
Variable r not set before use in line 7
Variable d not declared in line 8
Variable d not set before use in line 8
Variable area not set before use in line 10

Number of declared variables: 5
Number of declared integer variables: 1
Number of declared real variables: 4

Number of double declared variables: 2
More specifically:
r:3
circ:2
Number of double declared integer variables: 0
Number of double declared real variables: 2

Syntactically correct program
```