

Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών**Τμήμα Μηχανικών Η/Υ & Πληροφορικής****Πανεπιστήμιο Πατρών****Εαρινό Εξάμηνο 2024****Διδάσκοντες: Ι. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας****Εργαστηριακή Άσκηση**

Σκοπός της εργαστηριακής άσκησης είναι να εξοικειωθείτε με την περιγραφή μιας γλώσσας σε μορφή BNF, βασικές έννοιες των μεταγλωττιστών και τελικά να υλοποιήσετε δύο από τα τμήματα ενός μεταγλωττιστή, έναν συντακτικό και ένα λεκτικό αναλυτή με τη χρήση των εργαλείων Bison και Flex.

ΔΙΑΔΙΚΑΣΤΙΚΑ**Παραδοτέα**

- ❖ **Γραπτή Αναφορά** που περιλαμβάνει:
 - Τα στοιχεία των μελών της ομάδας (Ονοματεπώνυμο, ΑΜ, Έτος, e-mail)
 - Την περιγραφή της γραμματικής της γλώσσας σε BNF.
 - Τα **ΤΕΛΙΚΑ** αρχεία περιγραφής της γλώσσας, τα οποία δίνονται ως **είσοδος** στα Flex και Bison
 - Screenshots παραδειγμάτων εφαρμογής (τουλάχιστον ένα παράδειγμα για κάθε ερώτημα, για επιτυχημένη και ανεπιτυχή ανάλυση)
 - Σχόλια - Παραδοχές που έγιναν για την ανάπτυξη της εργασίας
- ❖ Συμπίεσμένα σε **ένα αρχείο (zip)** τα:
 - Την παραπάνω γραπτή αναφορά.
 - Τα **ΤΕΛΙΚΑ** αρχεία εισόδου για το flex και το bison.
 - Τον **ΤΕΛΙΚΟ** κώδικα σε C (και το .h αρχείο) που παράγεται από τα δύο προγράμματα και ο εκτελέσιμος κώδικας του parser.
 - Τα **δοκιμαστικά αρχεία** που δόθηκαν σαν είσοδο στον parser για να ελεγχθεί η σωστή λειτουργία του.

Το αρχείο zip πρέπει να έχει όνομα τους αριθμούς μητρώου των ατόμων της ομάδας διαχωρισμένους με _ και από τον μικρότερο στο μεγαλύτερο (π.χ. 1000_1543_2788_3972.zip), και να υποβληθεί (ΥΠΟΧΡΕΩΤΙΚΑ) στο eclass του μαθήματος και στο μήνυμα να αναφέρονται τα ονοματεπώνυμα, το έτος και οι αντίστοιχοι αριθμοί μητρώου των μελών της ομάδας, καθώς και οι e-mail διευθύνσεις όλων των μελών της ομάδας

Διευκρινίσεις

- ❖ Για τη λειτουργία των εργαλείων Flex και Bison μπορείτε να βρείτε πληροφορίες στο eclass του μαθήματος.
- ❖ Η άσκηση θα γίνει σε ομάδες από 1 έως 4 άτομα.
- ❖ Η συμβολή της στον τελικό βαθμό είναι 30%.
- ❖ Η άσκηση είναι υποχρεωτική και πρέπει ο βαθμός της να είναι τουλάχιστον 5.
- ❖ Οι τελικές ημερομηνίες παράδοσης θα καθοριστούν αναλόγως με τις ημερομηνίες γραπτών εξετάσεων περιόδου Ιουνίου και Σεπτεμβρίου αντίστοιχα. Αναλόγως θα καθοριστεί και η ημερομηνία της προφορικής εξέτασης της εργασίας.
- ❖ Τυχόν απορίες ή υποδείξεις αναρτώνται **ΑΠΟΚΛΕΙΣΤΙΚΑ** στο χώρο “Συζητήσεις” στην σελίδα του μαθήματος στο eclass

(<https://eclass.upatras.gr/modules/forum/?course=CEID1091>)

ΠΕΡΙΓΡΑΦΗ ΑΣΚΗΣΗΣ

Η γλώσσα που θα αναλυθεί είναι μια φανταστική αντικειμενοστραφής γλώσσα προγραμματισμού. Τα προγράμματα της γλώσσας οργανώνονται σε *classes*, όπως παρουσιάζεται στις προδιαγραφές της ακολούθως.

Προδιαγραφές γλώσσας

- Κάθε πρόγραμμα της γλώσσας αποτελείται από μια ή περισσότερες *classes*. Μια *class* μπορεί να είναι εμφωλευμένη εντός άλλης *class*.
- Τα αναγνωριστικά (*identifiers*) της γλώσσας απαρτίζονται από γράμματα (**A...Z, a...z**), ψηφία (**0...9**) και τον ειδικό χαρακτήρα κάτω παύλας **_**. Ένα αναγνωριστικό δεν μπορεί να ξεκινά από ψηφίο. Η γλώσσα είναι *case sensitive*.
- Κάθε *class* ακολουθεί την παρακάτω μορφή:

```
public class <όνομα_class> {  
    προαιρετική δήλωση μεταβλητών  
    προαιρετική δήλωση μεθόδων  
}
```

Το όνομα κάθε *class* ξεκινά με κεφαλαίο γράμμα.

Για παράδειγμα:

```
public class A {  
    public void method1() {  
        out.println("Hello world!");  
    }  
}
```

- Τα ακόλουθα αναγνωριστικά της γλώσσας είναι δεσμευμένα και δεν μπορούν να χρησιμοποιηθούν ως κοινά αναγνωριστικά: **int**, **char**, **double**, **boolean**, **String**, **class**, **new**, **return**, **void**, **if**, **else**, **while**, **do**, **for**, **switch**, **case**, **default**, **break**, **true**, **false**, **public**, **private**.
- Οι βασικοί (*primitive*) τύποι δεδομένων που υποστηρίζει η γλώσσα είναι οι **int**, **char**, **double** και **boolean**.
 - Ο τύπος δεδομένων **int** χρησιμοποιείται για τη δήλωση/δημιουργία μεταβλητών ακεραίου τύπου.
 - Ο τύπος δεδομένων **char** χρησιμοποιείται για τη δήλωση/δημιουργία μεταβλητών που αποτελούνται από έναν απλό ή ειδικό χαρακτήρα (π.χ. χαρακτήρας νέας γραμμής) που περικλείεται από **μονά** εισαγωγικά. Για παράδειγμα **'a'**, **'b'**, **'9'**, **'\n'**.
 - Ο τύπος δεδομένων **double** χρησιμοποιείται για τη δήλωση/δημιουργία μεταβλητών κινητής υποδιαστολής. Στο τέλος κάθε τιμής αυτού του τύπου θα πρέπει να υπάρχει ο χαρακτήρας **d**, π.χ. 5.99d, 6.0d, 123545.23436d.
 - Ο τύπος δεδομένων **boolean** χρησιμοποιείται για τη δήλωση/δημιουργία μεταβλητών λογικού τύπου και μπορεί να πάρει αποκλειστικά μόνο τις τιμές

true ή false.

- Η γλώσσα υποστηρίζει επιπλέον το μη-βασικό (non-primitive) τύπο δεδομένων **String**, ο οποίος χρησιμοποιείται για τη δήλωση μιας ακολουθίας χαρακτήρων που περικλείεται από διπλά εισαγωγικά.
Για παράδειγμα:

```
"Hello123$", "This is a valid str1ng", "this is \'valid\' too"
```

- Η **δήλωση μεταβλητών** ακολουθεί την εξής μορφή:

```
[<modifier>] <τύπος_δεδομένων> <όνομα_μεταβλητής>;
```

ενώ οι επιτρεπόμενοι τύποι δεδομένων είναι αυτοί που παρουσιάστηκαν προηγουμένως. Προαιρετικά, μια μεταβλητή μπορεί να δηλωθεί ως public ή private (όρος modifier).
Για παράδειγμα:

```
int var1; private double var2;
```

- Η **δημιουργία ενός αντικειμένου μιας class** ακολουθεί την εξής μορφή:

```
<όνομα_class> <όνομα_αντικειμένου> = new <όνομα_class>();
```

Η **πρόσβαση σε μέλη** (μεταβλητή, μέθοδος) μιας class ακολουθεί την εξής μορφή:

```
<όνομα_αντικειμένου>.<όνομα_μέλους_class>;
```

- Ο **ορισμός των μεθόδων** ακολουθεί τον τρόπο ορισμού που ισχύει στη Java. Στο σώμα οποιασδήποτε μεθόδου υπάρχει αρχικά προαιρετική δήλωση μεταβλητών και στη συνέχεια ακολουθούν οι εντολές (οι επιτρεπόμενες εντολές της γλώσσας παρουσιάζονται αργότερα). Μια μέθοδος μπορεί να είναι public ή private. Μια μέθοδος μπορεί να επιστρέφει ή όχι τιμή. Μια μέθοδος μπορεί να λαμβάνει κανένα, ένα ή περισσότερα ορίσματα.

Παρουσιάστε και τεκμηριώστε πλήρως τον ορισμό μεθόδου στη γραμματική σας.

- Μια μεταβλητή ή μια μέθοδος που είναι public είναι προσβάσιμη από αντικείμενα όλων των classes. Μια μεταβλητή ή μια μέθοδος που είναι private είναι προσβάσιμη μόνο από αντικείμενα της class που έχει οριστεί.
- Οι λοιπές επιτρεπόμενες εντολές προγράμματος της γλώσσας διακρίνονται σε:
 - *Εντολές ανάθεσης*

Έχουν τη μορφή **<μεταβλητή> = <έκφραση>;**, όπου η έκφραση μπορεί να είναι ένα κυριολεκτικό ή μια σύνθετη παράσταση από συνδυασμό κυριολεκτικών, μεταβλητών και παρενθέσεων, κλήση μεθόδου ή δημιουργία αντικειμένου class. Η κλήση μεθόδου ακολουθεί τη λογική της Java. Επιπλέον, η σύνθετη παράσταση μπορεί να είναι οποιαδήποτε αριθμητική παράσταση που περιλαμβάνει τις πράξεις +, -, *, /.

- ο *Εντολές βρόχου*

Υπάρχουν 2 είδη εντολών βρόχου. Το πρώτο είδος έχει την ακόλουθη μορφή:

```
do {  
    //εντολές προγράμματος  
}  
while (συνθήκη);
```

Όπου **συνθήκη** είναι οποιαδήποτε λογική έκφραση που περιέχει τους συγκριτικούς τελεστές **>**, **<**, **==**, **!=** και τους λογικούς **&&** και **||**.

Το δεύτερο είδος έχει την ακόλουθη μορφή:

```
for (έκφραση1; έκφραση2; έκφραση3) {  
    //εντολές προγράμματος  
}
```

όπου: η έκφραση1 αφορά την αρχικοποίηση της μεταβλητής του βρόχου (π.χ. `int i = 0`), η έκφραση2 είναι μια λογική έκφραση που ορίζει τη συνθήκη εκτέλεσης του βρόχου (π.χ. `i < 10`), και η έκφραση3 είναι μια εντολή ανάθεσης που τροποποιεί την τιμή της μεταβλητής του βρόχου (π.χ. `i = i + 15`).

- ο *Εντολές ελέγχου*

Υπάρχουν δυο είδη εντολών ελέγχου. Το πρώτο είδος έχει την ακόλουθη μορφή:

```
if (συνθήκη1) {  
    // εντολές προγράμματος  
} else if (συνθήκη2) {  
    // εντολές προγράμματος  
} else if (συνθήκη3) {  
    // εντολές προγράμματος  
}  
...  
else {  
    // εντολές προγράμματος  
}
```

Τα `else if` και `else` μπλοκ είναι προαιρετικά.

Το δεύτερο είδος έχει την ακόλουθη μορφή:

```
switch (έκφραση) {  
  case έκφραση:  
    // εντολές προγράμματος  
  case έκφραση:  
    // εντολές προγράμματος  
  ...  
  default:  
    // εντολές προγράμματος  
}
```

Η επιλογή default είναι προαιρετική.

- *Εντολή εκτύπωσης στην οθόνη:*

Έχει τη μορφή **out.print**("κείμενο"[var]);

(Δηλαδή κείμενο εντός "" κι αμέσως μετά προαιρετικά μεταβλητή/ές)

- *Εντολή επιστροφής από μέθοδο:*

Η εντολή **return** χρησιμοποιείται για την επιστροφή από κάποια μέθοδο. Όταν πρόκειται για μέθοδο που επιστρέφει κάποιον τύπο, το return ακολουθείται από έκφραση και ;.

- *Εντολή τερματισμού βρόχου:*

Έχει τη μορφή **break**;

- Σε οποιοδήποτε σημείο του κώδικα υποστηρίζονται τα σχόλια της γλώσσας. Αυτά μπορούν να είναι δυο μορφών:
 - Σχόλια μιας γραμμής: Τα σχόλια μιας γραμμής ξεκινούν με την ακολουθία **//**. Οτιδήποτε ακολουθεί τις **//** μέχρι την αλλαγή γραμμής αποτελεί σχόλιο και αγνοείται.
 - Σχόλια πολλαπλών γραμμών: Τα σχόλια πολλαπλών γραμμών ορίζονται με την ακολουθία **/* ... */**. Οτιδήποτε περιέχεται εντός των **/* ... */** (και οι αλλαγές γραμμής) αποτελεί σχόλιο και αγνοείται.
- Εάν δεν αναφέρεται ρητά κάτι διαφορετικό, τα κενά και οι χαρακτήρες αλλαγής γραμμής δεν αποτελούν μέρος της γλώσσας και δεν παίζουν κανένα ρόλο στη σύνταξη.

Ερωτήματα

1. (60%)

- a. Δώστε σε BNF το συντακτικό ορισμό της γραμματικής της γλώσσας.
- b. Χρησιμοποιώντας τα προγράμματα Flex και Bison, υλοποιήστε έναν λεξικό και συντακτικό αναλυτή, ο οποίος θα παίρνει ως είσοδο ένα αρχείο γραμμένο στη ψευδογλώσσα που περιγράφηκε παραπάνω και θα ελέγχει σε ένα πέρασμα αν το πρόγραμμα είναι συντακτικά ορθό. Το πρόγραμμά σας θα καλείται από τη γραμμή εντολών ως εξής:

prompt> myParser.exe file_name

και θα επιστρέφει το ίδιο το πρόγραμμα στην οθόνη και ένα διαγνωστικό μήνυμα για το αν ήταν ορθώς γραμμένο, ή κατάλληλο μήνυμα σφάλματος (πρέπει να φαίνεται η γραμμή όπου υπάρχει το σφάλμα και η ανάλυση να διακόπτεται εκεί).

2. (10%)

Τροποποιήστε τον αναλυτή σας, ώστε

- a. να αναγνωρίζεται ανάθεση τιμής κατά τη δήλωση μιας μεταβλητής. Αυτή είναι της μορφής [*<modifier>*]**<τύπος_δεδομένων> <μεταβλητή> = <έκφραση>**;
Για παράδειγμα:

int var1 = 12;

public String my_str = "H3ll0";

- b. να αναγνωρίζεται η δήλωση πολλαπλών μεταβλητών ίδιου τύπου σε μια γραμμή. Αυτή είναι της μορφής

<τύπος_δεδομένων> <μεταβλητή> = <έκφραση> , <μεταβλητή> = <έκφραση>, <μεταβλητή> = <έκφραση>, ...;

ή

<τύπος_δεδομένων> <μεταβλητή> , <μεταβλητή> , <μεταβλητή>, ...;

Για παράδειγμα:

int x = 15, y = 16, z = 17;

double x, y, z;

Παρουσιάστε την επικαιροποιημένη BNF.

3. (20%)

- a. Κάνετε τις απαραίτητες αλλαγές στον κώδικά σας, ώστε να γίνεται έλεγχος σωστής δήλωσης των μεταβλητών που χρησιμοποιούνται οπουδήποτε στο πρόγραμμα. Επιπλέον, να ελέγχεται ότι όπου καλούνται μέθοδοι, αυτές έχουν οριστεί.
Διαφορετικά, η διαδικασία της ανάλυσης να διακόπτεται με εμφάνιση σχετικού ανά περίπτωση σφάλματος.
- b. Κάνετε τις απαραίτητες αλλαγές στον κώδικά σας, ώστε να γίνεται έλεγχος εμβέλειας κατά τη χρήση των μεταβλητών και των μεθόδων.

- c. Επεκτείνετε τον κώδικά σας, ώστε, εάν σε μια μεταβλητή γίνεται ανάθεση αριθμητικής έκφρασης, να υπολογίζεται το αποτέλεσμα της έκφρασης - ακολουθώντας τους κανόνες προτεραιότητας - και να εμφανίζεται στην οθόνη επιπλέον μήνυμα με το όνομα της μεταβλητής και της τιμής που της έχει ανατεθεί.

Στην αριθμητική έκφραση μπορούν να συμμετέχουν αριθμοί, παρενθέσεις, και μεταβλητές που θα πρέπει να έχουν λάβει τιμές σε προηγούμενες εντολές - αλλιώς να διακόπτεται η διαδικασία ανάλυσης με εμφάνιση κατάλληλου σφάλματος.

Σημείωση: εάν υλοποιηθεί το παρόν ερώτημα, στα παραδοτέα σας να περιλαμβάνονται τα δοκιμαστικά αρχεία που θα επιβεβαιώνουν τη λειτουργία αυτή.

4. (10%)

Κάνετε τις απαραίτητες αλλαγές στο πρόγραμμά σας, ώστε σε περίπτωση σφάλματος να μην σταματά η εκτέλεση του προγράμματος, αλλά να συνεχίζει και να αναγνωρίζονται όλα τα πιθανά σφάλματα μέχρι το τέλος του αρχείου εισόδου.

Σημείωση: εάν υλοποιηθεί το παρόν ερώτημα, στα παραδοτέα σας να περιλαμβάνονται τόσο τα αρχεία που αφορούν την περίπτωση διακοπής της ανάλυσης κατά τον εντοπισμό σφάλματος, όσο και τα αρχεία που αφορούν την εύρεση όλων των σφαλμάτων.