

## Προγραμματιστική άσκηση:

### Η γλώσσα προγραμματισμού EEL

Η EEL (Early Experimental Language) είναι μια μικρή γλώσσα προγραμματισμού. Παρόλο που οι προγραμματιστικές της ικανότητες είναι μικρές, η εκπαιδευτική αυτή γλώσσα περιέχει πλούσια στοιχεία και η κατασκευή του μεταγλωττιστή της έχει να παρουσιάσει αρκετό ενδιαφέρον, αφού περιέχονται σε αυτήν πολλές εντολές που χρησιμοποιούνται από άλλες γλώσσες, καθώς και κάποιες πρωτότυπες. Η EEL υποστηρίζει συναρτήσεις και διαδικασίες, μετάδοση παραμέτρων με αναφορά και τιμή, αναδρομικές κλήσεις, και άλλες ενδιαφέρουσες δομές. Επίσης, επιτρέπει φώλιασμα στη δήλωση συναρτήσεων και διαδικασιών κάτι που λίγες γλώσσες υποστηρίζουν (το υποστηρίζει η Pascal, δεν το υποστηρίζει η C).

Από την άλλη όμως πλευρά, η EEL δεν υποστηρίζει βασικά προγραμματιστικά εργαλεία όπως η δομή *for*, ή τύπους δεδομένων όπως οι πραγματικοί αριθμοί και οι συμβολοσειρές. Οι παραλήψεις αυτές έχουν γίνει ώστε να απλουστευτεί η διαδικασία κατασκευής του μεταγλωττιστή, μία απλούστευση όμως που έχει να κάνει μόνο με τη μείωση των γραμμών κώδικα και όχι με τη δυσκολία κατασκευής του.

### Λεκτικές μονάδες

Το αλφάβητο της EEL αποτελείται από:

- τα μικρά και κεφαλαία γράμματα της λατινικής αλφαβήτου («A»,...,«Z» και «a»,...,«z»),
- τα αριθμητικά ψηφία («0»,...,«9»),
- τα σύμβολα των αριθμητικών πράξεων («+», «-», «\*», «/»),
- τους τελεστές συσχέτισης «<», «>», «=», «<=», «>=», «<>»,
- το σύμβολο ανάθεσης «:=»,
- τους διαχωριστές («;», «,», «:»)
- καθώς και τα σύμβολα ομαδοποίησης («(», «)», «[», «]»)
- και διαχωρισμού σχολίων (~~«/\*», «\*/», «//»~~) («/\*», «\*/», «//»).

Τα σύμβολα «[» και «]» χρησιμοποιούνται στις λογικές παραστάσεις όπως τα σύμβολα «(» και «)» στις αριθμητικές παραστάσεις.

Μερικές λέξεις είναι δεσμευμένες:

**program, endprogram**

**declare, enddeclare**

**if then else endif**

**while, endwhile**

**repeat, endrepeat, exit**

**switch, case, endswitch**

**forcase, when, endforcase**

**procedure, endprocedure, function, endfunction, call, return, in, inout**

**and, or, not, true, false**

**input, print**

Οι λέξεις αυτές δεν μπορούν να χρησιμοποιηθούν ως μεταβλητές. Οι σταθερές της γλώσσας είναι ακέραιες σταθερές που αποτελούνται από προαιρετικό πρόσημο και από μία ακολουθία αριθμητικών ψηφίων. Υπάρχουν και οι σταθερές **true** και **false**.

Τα αναγνωριστικά της γλώσσας είναι συμβολοσειρές που αποτελούνται από γράμματα και ψηφία, αρχίζοντας όμως από γράμμα. Ο μεταγλωττιστής λαμβάνει υπόψη του μόνο τα τριάντα πρώτα γράμματα. Οι λευκοί χαρακτήρες (tab, space, return) αγνοούνται και μπορούν να χρησιμοποιηθούν με οποιονδήποτε τρόπο χωρίς να επηρεάζεται η λειτουργία του μεταγλωττιστή, αρκεί βέβαια να μην βρίσκονται μέσα σε δεσμευμένες λέξεις, αναγνωριστικά, σταθερές. Το ίδιο ισχύει και για τα σχόλια, τα οποία πρέπει να βρίσκονται μέσα στα σύμβολα /\* και \*/ ή να βρίσκονται μετά το σύμβολο // και ως το τέλος της γραμμής.

### **Μορφή προγράμματος**

**program id**

*declarations*

*subprograms*

*statements*

**endprogram**

## Τύποι και δηλώσεις μεταβλητών

Ο μοναδικός τύπος δεδομένων που υποστηρίζει η EEL είναι οι ακέραιοι αριθμοί. Οι ακέραιοι αριθμοί πρέπει να έχουν τιμές από -32767 έως 32767. Η δήλωση γίνεται με την εντολή **declare**. Ακολουθούν τα ονόματα των αναγνωριστικών χωρίς καμία άλλη δήλωση, αφού γνωρίζουμε ότι πρόκειται για ακέραιες μεταβλητές και χωρίς να είναι αναγκαίο να βρίσκονται στην ίδια γραμμή. Οι μεταβλητές χωρίζονται μεταξύ τους με κόμματα. Το τέλος της δήλωσης των μεταβλητών γίνεται με την εντολή **enddeclare**.

## Τελεστές και εκφράσεις

Η προτεραιότητα των τελεστών από τη μεγαλύτερη στη μικρότερη είναι:

- (1) Μοναδιαίοι λογικοί: «not»
- (2) Πολλαπλασιαστικοί: «\*», «/»
- (3) Μοναδιαίοι προσθετικοί: «+», «-»
- (4) Δυαδικοί προσθετικοί: «+», «-»
- (5) Σχεσιακοί «=», «<», «>», «<>», «<=», «>=»
- (6) Λογικό «and»,
- (7) Λογικό «or»

## Δομές της γλώσσας

### Εκχώρησης

*Id := expression*

Χρησιμοποιείται για την ανάθεση της τιμής μίας μεταβλητής ή μίας σταθεράς, ή μίας έκφρασης σε μία μεταβλητή.

### Απόφασης if

```
if condition then
    statements
[else
    statements]
endif
```

Η εντολή απόφασης **if** εκτιμάει εάν ισχύει η συνθήκη *condition* και εάν πράγματι ισχύει, τότε εκτελούνται οι εντολές που ακολουθούν το **then** έως ότου συναντηθεί **else** ή **endif**. Το **else** δεν αποτελεί υποχρεωτικό τμήμα της εντολής και γι' αυτό βρίσκεται σε αγκύλη. Οι εντολές που το ακολουθούν εκτελούνται εάν η συνθήκη *condition* δεν ισχύει. Το **endif** είναι υποχρεωτικό τμήμα της εντολής.

#### Επανάληψης repeat

```
repeat  
    statements  
endrepeat
```

Η εντολή επανάληψης **repeat** επαναλαμβάνει συνεχώς τις εντολές που βρίσκονται ανάμεσα στο **repeat** και στο **endrepeat** έως ότου εκτελεστεί η **exit**. Με την εντολή **exit** η εκτέλεση μεταφέρεται έξω από τον βρόχο.

#### Επιστροφής

```
return expression
```

Χρησιμοποιείται μέσα σε συναρτήσεις για να επιστραφεί το αποτέλεσμα της συνάρτησης.

#### Επανάληψης while

```
while condition  
    statements  
endwhile
```

Η εντολή επανάληψης **while** επαναλαμβάνει τις εντολές που βρίσκονται ανάμεσα στο **while** και στο **endwhile** για όσο ισχύει η συνθήκη *condition*. Αν στον πρώτο έλεγχο, η συνθήκης δεν είναι αληθής, τότε οι εντολές *statements* δεν εκτελούνται ποτέ.

#### Απόφασης switch

```
switch expression  
    case expression : statements  
    (case expression : statements)*  
endswitch
```

Η δομή απόφασης **switch** ελέγχει εάν η έκφραση *expression* που υπάρχει ακριβώς μετά από τη δεσμευμένη λέξη **switch** ισούται με κάποια από τις υπόλοιπες *expression* που βρίσκονται μετά τα **case**, εξετάζοντας τις κατά σειρά. Για την πρώτη από αυτές που ισχύει ότι είναι ίση με την αρχική *expression* εκτελούνται οι εντολές που ακολουθούν το σύμβολο ":". Αφού εκτελεστεί κάποια από τις *statements* τότε ο έλεγχος μεταβαίνει έξω από τη δομή **switch**. Εάν καμία από τις *statements* δεν εκτελεστεί, τότε πάλι ο έλεγχος μεταβαίνει έξω από τη δομή **switch**.

### Επανάληψης **forcase**

```
forcase  
    when condition : statements  
    (when condition : statements)*  
endforcase
```

Η δομή επανάληψης **forcase** ελέγχει τις *condition* που βρίσκονται μετά τα **when**, εξετάζοντας τις κατά σειρά. Για κάθε μία από αυτές που ισχύει εκτελούνται οι *statements* που ακολουθούν το σύμβολο ":". Θα εξεταστούν όλες οι *condition* και θα εκτελεστούν όλες οι *statements* των οποίων οι *condition* ισχύουν. Αφότου εξετατούν όλες οι **when** ο έλεγχος μεταβαίνει έξω από τη δομή **forcase** εάν καμία από τις *statements* δεν έχει εκτελεστεί ή μεταβαίνει στην αρχή της **forcase** εάν έστω και μία από τις *statements* έχει εκτελεστεί.

### Εξόδου

```
print expression
```

Εμφανίζει στην οθόνη το αποτέλεσμα της αποτίμησης του *expression*

### Εισόδου

```
input id
```

Ζητάει από τον χρήστη να δώσει μία τιμή μέσα από το πληκτρολόγιο

## Υποπρογράμματα

Η EEL υποστηρίζει και συναρτήσεις και διαδικασίες.

Διαδικασία:

```
procedure id (formal_pars)  
    declarations  
    subprograms  
    statements  
endprocedure
```

Συνάρτηση:

```
function id (formal_pars)  
    declarations  
    subprograms
```

*statements*

### ***endfunction***

Η «*formal\_pars*» είναι η λίστα των τυπικών παραμέτρων. Οι διαδικασίες και οι συναρτήσεις μπορούν να φωλιάσουν η μία μέσα στην άλλη και οι κανόνες εμβέλειας είναι όπως της PASCAL. Η επιστροφή της τιμής μιας συνάρτησης γίνεται με την **return**

Η κλήση μιας διαδικασίας γίνεται με την **call**

### ***call id (actual\_parameters)***

ενώ όταν πρόκειται για συνάρτηση, αυτή λαμβάνει μέρος στις αριθμητικές παραστάσεις σαν τελούμενο. π.χ.

$$D = a + f(\text{in } x)$$

όπου *f* η συνάρτηση και *x* παράμετρος που περνάει με τιμή.

## **Μετάδοση παραμέτρων**

Η EEL υποστηρίζει δύο τρόπους μετάδοσης παραμέτρων:

- με σταθερή τιμή. Δηλώνεται με τη λεκτική μονάδα **in**. Αλλαγές στην τιμή της δεν επιστρέφονται στον καλόν πρόγραμμα
- με αναφορά. Δηλώνεται με τη λεκτική μονάδα **inout**. Κάθε αλλαγή στη τιμή της μεταφέρεται και στο πρόγραμμα που κάλεσε τη διαδικασία ή τη συνάρτηση

Στην κλήση μίας συνάρτησης ή μίας διαδικασίας οι πραγματικοί παράμετροι συντάσσονται μετά από τις λέξεις κλειδιά **in** και **inout**, ανάλογα με το αν περνάνε με τιμή ή αναφορά.

## Παράδειγμα 1

```
program example1
  declare d,i,g,f enddeclare

  procedure two (in g)
    function three (in g, inout x, inout m)
      declare k, j enddeclare
      k:=g;
      j:=g;
      repeat
        if k>i then
          exit
        endif;
        j:=j*k;
        k:=k+g
      endrepeat;
      m:=j;
      return m+1;
      x:=7
    endfunction
    i:=three (in i+2, inout d, inout f)
  endprocedure

  function one (in g)
    call two(in g)
  endfunction

  i:=5;
  g:=1;
  call one (in g)
endprogram
```

## Παράδειγμα 2

```
program example2
  declare x,y,z enddeclare

  procedure p1(in x, inout z, inout v)
    declare w enddeclare

    procedure p2(inout z)
      declare q enddeclare

      procedure p3(inout a, inout b)
        declare k enddeclare
        if v<>0 then
          v:=z+b;
          a:=1
        else
          a:=v/b
        endif;
        k:=x
      endprocedure

      q:=y+w;
      z:=q*x;
      call p3(inout q, inout v)
    endprocedure

    if x<y then
      w:=x+y
    else
      w:=x*y
    endif;
    call p2(inout z)
  endprocedure

  x:=1;
  y:=2;
  call p1(in x+y, inout z)
endprogram
```



### Παράδειγμα 3

```
program example3
  declare a,b,c,d,e,x,y,px,py,temp enddeclare
  repeat
    if not [a<c and b<d] then
      exit
    endif;
    if a=e then
      c:=c+e;
    else
      repeat
        if not [a<=d] then
          exit
        endif;
        a:=a+b;
      endrepeat
    endif
  endrepeat;
  temp:=px;
  x:=1;
  y:=2;
endprogram
```