

## Προγραμματιστική άσκηση:

### Η γλώσσα προγραμματισμού Starlet

Η Starlet είναι μια μικρή γλώσσα προγραμματισμού φτιαγμένη με βάση τις ανάγκες της προγραμματιστικής άσκησης του μαθήματος. 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 Starlet υποστηρίζει συναρτήσεις, μετάδοση παραμέτρων με αναφορά, τιμή και αντιγραφή, αναδρομικές κλήσεις και άλλες ενδιαφέρουσες δομές. Επίσης, επιτρέπει φώλιασμα στη δήλωση συναρτήσεων κάτι που λίγες γλώσσες υποστηρίζουν (το υποστηρίζει η Pascal, δεν το υποστηρίζει η C).

Από την άλλη όμως πλευρά, η Starlet 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 ή την εκπαιδευτική αξία της άσκησης.

Παρακάτω παρουσιάζεται μία περιγραφή της γλώσσας:

#### Λεκτικές μονάδες

Το αλφάβητο της Starlet 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***  
***declarations***  
***if then else endif***  
***do while endwhile***  
***loopf, endloop, exit***  
***forcase, endforcase, incase, endincase, when, default, enddefault***  
***function, endfunction, return, in, inout, inandout***  
***and, or, not***  
***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.

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. Απαγορεύεται να ανοίξουν δύο φορές σχόλια, πριν τα πρώτα κλείσουν. Δεν υποστηρίζονται εμφωλευμένα σχόλια.

### **Μορφή προγράμματος**

***program id***  
***declarations***  
***subprograms***  
***statements***  
***endprogram***

## Τύποι και δηλώσεις μεταβλητών

Ο μοναδικός τύπος δεδομένων που υποστηρίζει η Starlet είναι οι ακέραιοι αριθμοί. Οι ακέραιοι αριθμοί πρέπει να έχουν τιμές από -32767 έως 32767. Η δήλωση γίνεται με την εντολή **declarations**. 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 αναγνωρίζεται με το ελληνικό ερωτηματικό. Επιτρέπεται να έχουμε περισσότερες των μία συνεχόμενες χρήσεις της **declarations**.

## 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

### Εκχώρηση

*Id := expression*

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

### Απόφαση 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.

#### Επανάληψη **while**

```
while (condition)  
    statements  
endwhile
```

Η εντολή επανάληψης **while** επαναλαμβάνει συνεχώς τις εντολές *statements* που βρίσκονται ανάμεσα στο **while** και στο **endwhile**, όσο η συνθήκη *condition* ισχύει. Αν την πρώτη φορά που θα αποτιμηθεί η *condition*, το αποτέλεσμα της αποτίμησης είναι ψευδές, τότε οι *statements* δεν εκτελούνται ποτέ.

#### Επανάληψη **do-while**

```
do  
    statements  
while (condition)
```

Η εντολή επανάληψης **do-while** επαναλαμβάνει συνεχώς τις εντολές *statements* που βρίσκονται ανάμεσα στο **do** και στο **while**, όσο η συνθήκη *condition* ισχύει. Οι *statements* εκτελούνται τουλάχιστον μία φορά, πριν αποτιμηθεί η *condition*.

#### Επανάληψη **loop**

```
loop  
    statements  
endloop
```

Η εντολή επανάληψης **loop** επαναλαμβάνει για πάντα τις εντολές *statements* που βρίσκονται ανάμεσα στο **loop** και στο **endloop**. Έξοδος από το βρόχο γίνεται όταν κληθεί η εντολή **exit**

#### Επανάληψη **forcase**

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

Η δομή επανάληψης **forcase** ελέγχει τις *condition* που βρίσκονται μετά τα **when**. Μόλις μία από αυτές βρεθεί αληθής, τότε εκτελούνται οι *statements* που ακολουθούν. Μετά ο έλεγχος μεταβαίνει έξω από την **forcase**. Αν καμία από τις **when** δεν ισχύει, τότε ο έλεγχος

μεταβαίνει στη **default** και εκτελούνται οι αντίστοιχες *statements*. Στη συνέχεια ο έλεγχος μεταβαίνει στην αρχή της **forcase**.

#### Επανάληψη incase

```
incase  
    (when (condition): statements)*  
endincase
```

Η δομή επανάληψης **incase** ελέγχει τις *condition* που βρίσκονται μετά τα **when**, εξετάζοντας τις κατά σειρά. Για κάθε μία από αυτές που η αντίστοιχη *condition* 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 **incase** εάν καμία από τις *statements* δεν έχει εκτελεστεί ή μεταβαίνει στην αρχή της **incase**, εάν έστω και μία από τις *statements* έχει εκτελεστεί.

#### Επιστροφή τιμής

```
return expression
```

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

#### Έξοδος

```
print expression
```

Εμφανίζει στην οθόνη το αποτέλεσμα της αποτίμησης του *expression*

#### Είσοδος

```
input id
```

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

### Υποπρογράμματα

Η Starlet υποστηρίζει συναρτήσεις.

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

Η «*formal\_pars*» είναι η λίστα των τυπικών παραμέτρων. 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 συνάρτησης, γίνεται από τις 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 Starlet υποστηρίζει τρεις τρόπους μετάδοσης παραμέτρων:

- με σταθερή τιμή. Δηλώνεται με τη λεκτική μονάδα **in**. Αλλαγές στην τιμή της δεν επιστρέφονται στο πρόγραμμα που κάλεσε τη συνάρτηση.
- με αναφορά. Δηλώνεται με τη λεκτική μονάδα **inout**. Κάθε αλλαγή στη τιμή της μεταφέρεται αμέσως στο πρόγραμμα που κάλεσε τη συνάρτηση.
- με αντιγραφή. Δηλώνεται με τη λεκτική μονάδα **inandout**. Κάθε αλλαγή στη τιμή της μεταφέρεται στο πρόγραμμα που κάλεσε τη συνάρτηση, όταν ολοκληρώνεται η εκτέλεση της συνάρτησης.

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

### Κατάληξη

Τα αρχεία της Starlet έχουν κατάληξη **.stl**

## Παράδειγμα 1

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

  function two (in g)
    function three (in g, inout x, inandout m)
      declare k, j;
      k:=g;
      j:=g;
      do
        if (k>i) then
          k:=k-1
        endif;
        j:=j*k;
        k:=k+g
      while (k<1);
      m:=j;
      return m+1;
      x:=7
    endfunction
    i:=three (in i+2, inout d, inandout f)
  endfunction

  function one (in g)
    g:=two(in g)
  endfunction

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

## Παράδειγμα 2

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

  function p1(in x, inout z, inout v)
    declare w;

    function p2(inout z)
      declare q;

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

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

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

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



### Παράδειγμα 3

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