



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Προγραμματισμός Ι

Ενότητα 5 : Υποπρογράμματα ΙΙ

Αλέξανδρος Τζάλλας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Προγραμματισμός Ι

Ενότητα 5 : Υποπρογράμματα ΙΙ

Αλέξανδρος Τζάλλας

Λέκτορας

Άρτα, 2015





Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





Χρηματοδότηση

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Σκοποί ενότητας

- Να περιγραφούν οι συντακτικοί κανόνες των συναρτήσεων.
- Να γίνει κατανοητή η αναγκαιότητα χρήσης των συναρτήσεων.



Περιεχόμενα ενότητας

- Συναρτήσεις
- Σύνταξη των συναρτήσεων
- Κλήση συνάρτησης
- Αναγκαιότητα χρήσης των συναρτήσεων



Συναρτήσεις - Functions_{1/3}

- Οι συναρτήσεις (functions) είναι ο δεύτερος τύπος υποπρογραμμάτων που υποστηρίζει η Pascal
- Είναι υποπρογράμματα που παράγουν σαν αποτέλεσμα μία μόνο τιμή
- Η Pascal διαθέτει μια σειρά ενσωματωμένων (standard) συναρτήσεων όπως η `sqr`, η `round` κλπ που μπορούν να χρησιμοποιηθούν χωρίς να δηλωθούν



Συναρτήσεις - Functions_{2/3}

- Εκτός από τις παραπάνω η Pascal παρέχει την δυνατότητα ορισμού νέων συναρτήσεων
- Οι συναρτήσεις δηλώνονται στο τμήμα δηλώσεων του προγράμματος
- Η σειρά με την οποία δηλώνονται έχει σημασία μόνο όταν μία συνάρτηση καλεί μία άλλη
- Σε αυτή τη περίπτωση η καλούμενη πρέπει να προηγείται



Συναρτήσεις - Functions_{3/3}

- Μια συνάρτηση (function) μπορεί να θεωρηθεί σαν μια ειδική μορφή διαδικασίας (procedure) που
 - παίρνει έναν αριθμό δεδομένων εισόδου
 - εκτελεί μια λειτουργία ή ακολουθία λειτουργιών
 - παράγει ένα αποτέλεσμα
- Οι συναρτήσεις συντάσσονται όπως και οι διαδικασίες:
 - στην επικεφαλίδα αντί για **procedure** ξεκινάμε με την λέξη function
 - επειδή η συνάρτηση δίνει ένα αποτέλεσμα, στον ορισμό της, εκτός από το όνομα και τις παραμέτρους δίνεται και ο τύπος δεδομένων του αποτελέσματος (integer, real, char, boolean)
 - το τμήμα δηλώσεων και το τμήμα προτάσεων ορίζονται όπως αυτά των προγραμμάτων της Pascal



Σύνταξη_{1/2}

- Η λίστα παραμέτρων είναι προαιρετική όπως και στις διαδικασίες.
- Περιλαμβάνει δύο τύπους παραμέτρων:
 - τις παραμέτρους τιμής
 - τις παραμέτρους μεταβλητής
- Ο τύπος δεδομένων του αποτελέσματος μπορεί να είναι οποιοσδήποτε απλός τύπος

```
function Όνομα Συνάρτησης(Παράμετροι) :
    Τύπος Δεδομένων;
var
    Μεταβλητές : τύποι μεταβλητών
begin
    εντολές συνάρτησης
end.
```

Παράδειγμα

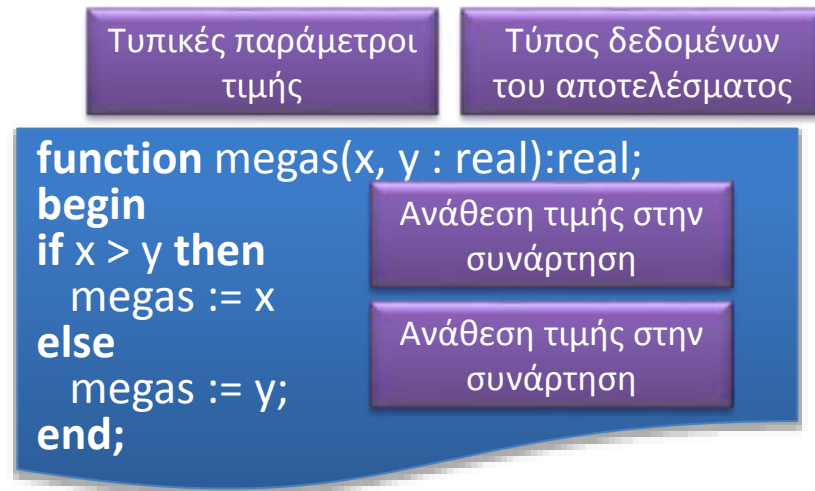
```
function exam(x : integer ; y : real) : boolean;

function test(x : real ; var k : char ; m :
integer) : char;
```



Σύνταξη_{2/2}

- Η συνάρτηση πρέπει να περιλαμβάνει τουλάχιστον μία πρόταση που να αναθέτει μία τιμή στο όνομα της συνάρτησης, διαφορετικά η κλήση της προκαλεί λάθος
- Σε κάθε κλήση και εκτέλεση της συνάρτησης, η τελευταία τιμή που ανατίθεται στο όνομα της συνάρτησης είναι και το αποτέλεσμα της





Κλήση Συνάρτησης

- Η κλήση μιας συνάρτησης διαφέρει από την κλήση μιας διαδικασίας
- Μία διαδικασία (procedure) καλείται από μια πρόταση στην οποία περιέχεται το όνομα της
- Μια συνάρτηση (function) καλείται μέσα από μία έκφραση στην οποία την τιμή συνεισφέρει και το αποτέλεσμα της συνάρτησης

```

program evresi_megaliterou_apo_duo;
var a, b, c : real;
{-----}
function max(x,y:real):real;
begin
If x > y then max := x
else max := y;
end;
{-----}
begin
writeln('Dwse duo arithmous');
readln(a,b);
c := max(a,b);
writeln('O megaliteros einai: ',c:5:2);
end.
    
```

Κλήση Συνάρτησης



Παρατηρήσεις

- Αν ένα υποπρόγραμμα υπολογίζει περισσότερα από ένα αποτελέσματα τότε είναι προτιμότερο να υλοποιηθεί με διαδικασία, αλλιώς (αν υπολογίζει μόνο ένα αποτέλεσμα) είναι προτιμότερο να υλοποιηθεί με συνάρτηση
- Δεν συνηθίζεται η χρήση **τυπικών παραμέτρων μεταβλητής** (με τη χρήση var) σε ορισμούς συναρτήσεων
- Αφού η συνάρτηση επιστρέφει μία τιμή στο σημείο όπου καλείται, μπορεί να χρησιμοποιηθεί & σαν πραγματική παράμετρος σε μιας κλήση υποπρογράμματος (διαδικασίας ή συνάρτησης)



Πότε χρησιμοποιούμε συναρτήσεις_{1/2}

- Αν το τμήμα του προγράμματος επιστρέφει περισσότερες από μία τιμές ή αλλάζει τιμές πραγματικών παραμέτρων τότε δεν χρησιμοποιούμε συνάρτηση
- Αν το τμήμα προγράμματος εκτελεί είσοδο δεδομένων ή έξοδο αποτελεσμάτων τότε δε χρησιμοποιούμε συνάρτηση
- Αν το τμήμα προγράμματος επιστρέφει μια τιμή και η τιμή αυτή είναι λογική (boolean) τότε χρησιμοποιούμε τη συνάρτηση



Πότε χρησιμοποιούμε συναρτήσεις_{2/2}

- Αν υπάρχει αμφιβολία είναι προτιμότερο να χρησιμοποιήσουμε μια διαδικασία (στη διαδικασία το ρόλο του ονόματος της συνάρτησης παίζει η μία παράμετρος μεταβλητής)
- Αν είναι δυνατόν να χρησιμοποιήσουμε και τις δύο δηλαδή διαδικασία ή συνάρτηση χρησιμοποιούμε αυτή με την οποία αισθανόμαστε πιο άνετα στην υλοποίηση



Παράδειγμα 1

- Πρόγραμμα που ελέγχει αν το άθροισμα των γωνιών ενός τριγώνου είναι 180° και αν είναι επιστρέφει την τιμή true.

Πότε τρεις γωνίες είναι γωνίες του ίδιου τριγώνου; Όταν αθροίζονται σε 180° μοίρες.

```

program athroisma_gwniwn_trigwnou;
var
    g1, g2, g3 : real;
function trigwno(gwnia1, gwnia2, gwnia3 : real):boolean;
begin
    trigwno := (gwnia1 + gwnia2 + gwnia3 - 180.0) = 0;
end;
{-----}
begin
    writeln('Δώσε τις τρεις γωνίες σε μοίρες');
    readln(g1, g2, g3);
    if trigwno(g1, g2, g3) then
        writeln('Οι τρεις γωνίες σχηματίζουν τρίγωνο')
    else
        writeln('Οι τρεις γωνίες δεν σχηματίζουν
    τρίγωνο');
end.
    
```

Κλήση Συνάρτησης



Παράδειγμα 2

✓ Πρόγραμμα που υπολογίζει την ακέραια δύναμη ενός πραγματικού αριθμού με την χρήση συνάρτησης

```

program ipologismos_dynamis;
var
    b, apotelesma : real;
    e : integer;

function dynami(basi:real ; ekthetis:integer):real;
var i: integer; d:real;
begin
    d:=1;
    for i:=1 to ekthetis do
        d:=d*basi;
    dynami := d;
end;
begin
    write('Δώσε τη βάση');
    read(b);
    write('Δώσε τον εκθέτη');
    read(e);
    apotelesma := dynami(b, e);
    write('Η δύναμη είναι: ',apotelesma:5:2);
end.
    
```

Κλήση Συνάρτησης



Παράδειγμα 3

Να γραφεί πρόγραμμα που να χρησιμοποιεί μια συνάρτηση η οποία να δίνει το τετράγωνο ενός αριθμού

```
program tetragwno_arith;  
var  
n: integer;  
function square(x: integer):integer;  
begin  
square:=x*x;  
end;  
begin  
n:=square(4);  
writeln('4 squared is', n);
```

```
program tetragwno_arith;  
var  
n,d: integer;  
function square(x: integer):integer;  
begin  
square:=x*x;  
end;  
begin  
writeln('dose arithmo ');  
readln(n);  
d:=0;  
d:=square(n);  
writeln('O ', n, ' arithmos squared is: ', d);  
end.
```



Παράδειγμα 4

Πρόγραμμα που
βρίσκει τον
μεγαλύτερο μεταξύ
τριών αριθμών

```

program megaliteros;
var
    ar1, ar2, ar3, z : integer;
function max(a, b, c : integer):integer;
begin
    if      a > b
    then if c > a
         then max := c
         else max := a
    else if c > b then max := c
         else max := b;
end;
begin
    writeln('Δώσε τρεις αριθμούς');
    readln(ar1, ar2, ar3);
    z := max(ar1, ar2, ar3);
    writeln('Μεγαλύτερος είναι ο ',z)
end.
    
```

Κλήση Συνάρτησης



Παράδειγμα 5

- Η Pascal δεν έχει ενσωματωμένη συνάρτηση για τον υπολογισμό της δύναμης.

x^n :

Αυτό μπορεί να γίνει με τη χρήση της πιο κάτω ιδιότητας των λογαρίθμων.

Αν $x = \alpha^v$

τότε $x = e^{(v \ln(\alpha))}$

```
program synartisi1;
var
x,n:real;
function power(x,n:real):real;
begin
power:=exp(n*ln(x));
end;
begin
write('Dose dyo arithmous...x, n: ');
readln(x,n);
writeln('x=',x:2:1,'n=',n:2:1);
writeln(x:2:1,'^',n:2:1,' = ',power(x,n):2:1);
end.
```

Κλήση Συνάρτησης



Παράδειγμα 6_{1/2}

Να γραφεί πρόγραμμα το οποίο να διαβάζει τις τιμές των συντελεστών a, β και γ , μιας δευτεροβάθμιας εξίσωσης. Το πρόγραμμα να υπολογίζει και να ελέγχει την διακρίνουσα αν είναι θετική ή όχι. Αν είναι θετική, $\Delta \geq 0$ να υπολογίζει τις πραγματικές τιμές x_1 και x_2 . Διαφορετικά, να τυπώνει το μήνυμα, “Μιγαδικές ρίζες”.

```

program diadikasia_synartisi;
var
a,b,c,x1,x2:real;
d:real;
function determinant(a,b,c:real):real;
begin
determinant:=b*b-4*a*c;
end;
procedure roots(a,b,c,d:real; var x1,x2:real);
begin
x1:=(-b+sqrt(d))/(2*a);
x2:=(-b-sqrt(d))/(2*a);
end;
begin
...

```



Παράδειγμα 6_{2/2}

Να γραφεί πρόγραμμα το οποίο να διαβάζει τις τιμές των συντελεστών α, β και γ , μιας δευτεροβάθμιας εξίσωσης. Το πρόγραμμα να υπολογίζει και να ελέγχει την διακρίνουσα αν είναι θετική ή όχι. Αν είναι θετική, $\Delta \geq 0$ να υπολογίζει τις πραγματικές τιμές x_1 και x_2 . Διαφορετικά, να τυπώνει το μήνυμα, “Μιγαδικές ρίζες”.

```

...
writeln(' Rizes tis deuterovathmias eksisosis');
write('Dose tous treis syntelestes tis eksisosis, a,b
kai c: ');
readln(a,b,c);
If determinant(a,b,c)>=0 then
    begin
        roots(a,b,c,determinant(a,b,c),x1,x2);
        writeln(' Oi Rizes einai ');
        writeln('x1= ',x1:2:2);
        writeln('x2= ',x2:2:2);
    end
else
    writeln(' Oi Rizes einai Migadiges ');
end.
    
```



Παράδειγμα 7

Ένας καθηγητής υπολογίζει τους βαθμούς του τριμήνου βάσει τριών διαγωνισμάτων. Η βαρύτητα του πρώτου διαγωνίσματος είναι 60% και των υπολοίπων δυο 20%. Το κάθε διαγώνισμα βαθμολογείται με μέγιστο βαθμό το 20. Να γράψετε πρόγραμμα που να διαβάζει τα αποτελέσματα των τριών διαγωνισμάτων και να υπολογίζει τον τελικό βαθμό του τριμήνου.

```
program synartiseis2;
const
w1=0.6;w2=0.2;w3=0.2;
var
test1,test2,test3:Real;
function finalgrade(test1,test2,test3:real):real;
begin
finalgrade:=w1*test1+w2*test2+w3*test3;
end;

begin
writeln('Dose tous vathmous twn triwn diagonismatwn');
Writeln;
write('Vathmos 1ou diagonismatos...');readln(test1);
write('Vathmos 2ou diagonismatos...');readln(test2);
write('Vathmos 3ou diagonismatos...');readln(test3);
write('O telikos vathmos tou triminou einai
',finalgrade(test1,test2,test3):1:1);
end.
```



Βιβλιογραφία

Βλαχάβας Ι. (1994). Η γλώσσα προγραμματισμού Pascal. Εκδόσεις Γαρταγάνης Διονύσιος.

Κάβουρας Ι.Κ. (1999). Δομημένος Προγραμματισμός με Pascal. Εκδόσεις Κλειδάριθμος.

Αλεβίζου Θ., & Καμπουρέλης Α. (1995). Μαθήματα Προγραμματισμού: Εισαγωγή με τη Γλώσσα Pascal. Εκδόσεις Παπασωτηρίου.

Cooper D. (1993). Oh! Pascal!, An Introduction to Computing, του. Εκδόσεις Norton.

Larry R.N. (1998). Advanced Programming in Pascal with Data Structures. Εκδόσεις Macmillan USA.

Τσελίκης Γ.Σ., Τσελίκας Ν.Δ. (2012). C: από τη Θεωρία στην Εφαρμογή (Β' Έκδοση). Εκδόσεις Παπασωτηρίου.

Aho A.V., Hopcroft J.E., & Ullman J.D. (1974). The design and analysis of computer algorithms. Εκδόσεις Addison Wesley.

Abelson H., Sussman G.J., Sussman J. (1985). Structure and Interpretation of Computer Programs, MIT Press, McGraw Hill Book Company.



Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Αλέξανδρος Τζάλλας.
Προγραμματισμός Ι.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/COMP111/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης
Άρτα, 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Τέλος Ενότητας

Υποπρογράμματα II



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης