



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών

Ενότητα 7: Υπορουτίνες

Καθ. Γιάννης Γαροφαλάκης

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Ορισμός

Αφαίρεση με χρήση υπορουτινών (subroutine abstraction) είναι η αντιστοίχιση ενός συνόλου εισόδων σε ένα σύνολο εξόδων που μπορεί να περιγραφεί φορμαλιστικά.

Η περιγραφή πρέπει να δείχνει πως σχετίζονται οι έξοδοι με τις εισόδους, αλλά δεν χρειάζεται να δείχνει τον τρόπο με τον οποίο υπολογίζονται οι έξοδοι.

Ο προγραμματιστής εστιάζει την προσοχή του στο τι γίνεται στο σημείο της κλήσης, και όχι στον τρόπο με τον οποίο γίνεται.



Είδη Υπορουτινών

- **Διαδικασία** (procedure)

Εκπληρώνει το έργο της, είτε εκχωρώντας τα αποτελέσματά της σε μία ή περισσότερες από τις παραμέτρους της, είτε αλλάζοντας το περιβάλλον της (τιμές μη-τοπικών μεταβλητών, ΠΑ), είτε κάνοντας και τα δύο.

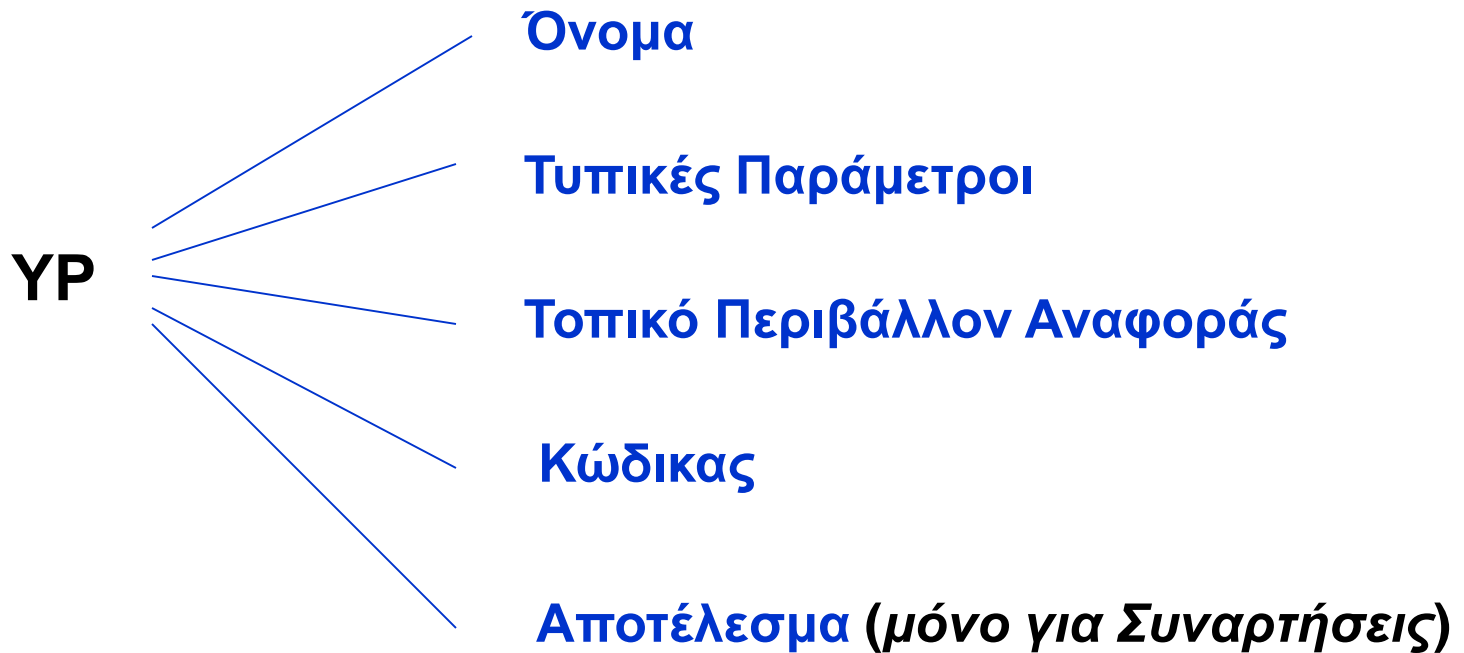
- **Συνάρτηση** (function)

Είναι Διαδικασία που, επιπλέον, επιστρέφει μία τιμή.



Συστατικά Υπορουτινών

Μία Υπορουτίνα (ΥΡ), περιλαμβάνει 4
(διαδικασίες), ή 5 (συναρτήσεις) **στοιχεία** :



Και άλλοι Ορισμοί

Τυπικές Παράμετροι: Δεν είναι μεταβλητές, αλλά μόνο ονόματα, που δείχνουν το ρόλο που θα παίξουν οι πραγματικές παράμετροι, όταν κληθεί η υπορουτίνα.

Πραγματικές Παράμετροι: Οι μεταβλητές και/ή οι εκφράσεις που παρέχονται σε μια υπορουτίνα, για να αντικαταστήσουν τις τυπικές παραμέτρους.



Παραδείγματα (1)

- Pascal:

```
PROCEDURE F(X: real; var Y: integer) [: real]
```

```
[FUNCTION]
```

```
VAR M: array[1..10] of real;
```

```
    N: integer
```

```
begin
```

```
...
```

```
end;
```



Παραδείγματα (2)

- C functions:

```
float power(float base, float exp)
```

```
{ ...  
  ... }
```

Κλήση:
`x = power(10.0, x)`

- C procedures:

```
void sort(int x[], int a)
```

```
{ ...  
  ... }
```

Κλήση:
`sort(scores, 100)`



Σχεδιαστικά Θέματα (1)

- **Λίστα Παραμέτρων ή Προσδιορισμός Παραμέτρων**

Εκτός από το **όνομα** των παραμέτρων, περιλαμβάνει τον **τύπο** και τον **τρόπο** χρήσης. Π.χ.

```
PROCEDURE F( X: real; var Y: integer)
```

- **Αντιστοιχία Τυπικών – Πραγματικών Παραμέτρων**

Με βάση τη σειρά αναγραφής στον ορισμό της υπορουτίνας.

Εξαίρεση: Η **Ada** επιτρέπει την ανατροπή της σειράς. Π.χ.

Ορισμός: **F(A, B)**. Κλήση: **F(B->100, A->10)**.



Σχεδιαστικά Θέματα (2)

- Είδος τιμών που επιστρέφει μια Συνάρτηση

FORTRAN, ALGOL60: Απλοί βαθμωτοί ΤΔ (real, int, bool)

PL/1: Βαθμωτοί και αλφαριθμητικά και pointers

Pascal: βαθμωτοί και pointers

C, Ada: Όλους τους ΤΔ.

Στις περισσότερες ΓΠ, πρέπει να εκχωρηθεί τιμή στο όνομα της function, πριν το τέλος της περιγραφής της. Π.χ. **Pascal:**

```
function f(x: integer): integer
begin
  if x<=1 then
    f:=1
  else
    f:=x*f(x-1)
end;
```



Υπολογισμός και Μεταβίβαση Παραμέτρων

- **Υπολογισμός Παραμέτρων (parameter evaluation)**

Διεργασία κατά την οποία κάθε πραγματική παράμετρος αναγνωρίζεται ότι συνδέεται με την αντίστοιχη τυπική παράμετρο, και μετά υπολογίζεται.

- **Μεταβίβαση Παραμέτρων (parameter passing)**

Ο τρόπος με τον οποίο η υπολογισμένη πραγματική παράμετρος μεταφέρεται (συνδέεται) στην υπορουτίνα.

- **Κλήση με Τιμή (call by value)**
- **Κλήση με Αναφορά (call by reference)**
- **Κλήση με Τιμή - Αποτέλεσμα (call by value – result)**
- **Κλήση με Όνομα (call by name)**
-



Κλήση με Τιμή (1)

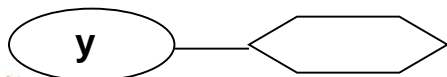
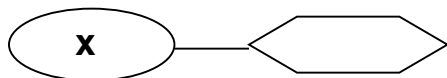
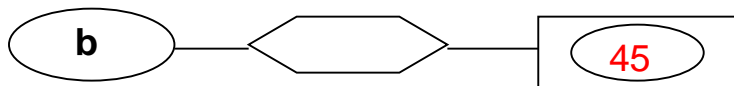
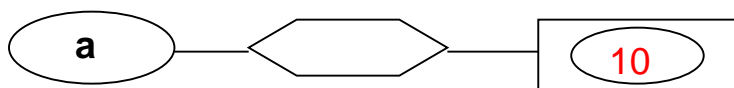
- Η πραγματική παράμετρος αποαναφοροποιείται και επιστρέφει μία τιμή, η οποία αντιγράφεται σε μια νέα θέση μνήμης, με την οποία συνδέεται το όνομα της τυπικής παραμέτρου. Δηλαδή, πρακτικά δημιουργείται μια νέα (τοπική) μεταβλητή.
- **C, C++, Pascal** (default)
- *Πλεονέκτημα*: Η υπορουτίνα μόνο διαβάζει την πραγματική παράμετρο, δεν έχει πρόσβαση για να την αλλάξει.
- *Μειονέκτημα*: Διπλασιασμός χρησιμοποιούμενης μνήμης.



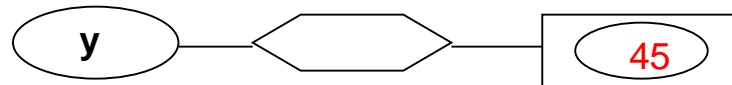
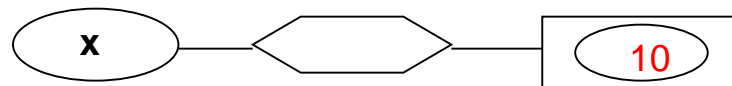
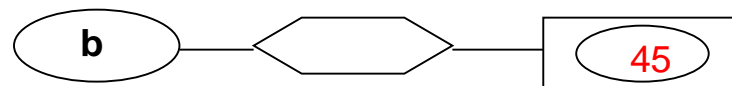
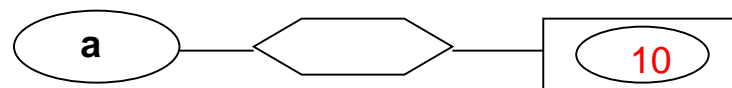
Κλήση με Τιμή (2)

Σχηματικά:

Ορισμός: Procedure $P(x, y)$



Κλήση: $P(a, b)$



Κλήση με Αναφορά (1)

- Αν η πραγματική παράμετρος χρησιμοποιεί μνήμη (π.χ. είναι μεταβλητή), αυτή η μνήμη συνδέεται με την τυπική παράμετρο, όταν γίνεται κλήση της υπορουτίνας. Δηλαδή, πρακτικά η τυπική παράμετρος γίνεται pointer στην πραγμ. παράμετρο.
- **FORTRAN, PL/1, Pascal** (με **VAR** στις τυπικές παραμέτρους)
- **C, C++** με χρήση pointers: `void A(int *f)` κλήση: `A(&x)`
- Κλήση με πραγματικές παραμέτρους σταθερές:

`P(var x)` Κλήση: `P(10)`

Αν αλλάζει η τιμή του *x* στη διαδικασία, η θέση μνήμης του, λέγεται *ανώνυμη μεταβλητή*.

- *Πλεονεκτήματα*: Απόδοση, γρήγορη αλλαγή τιμής της πραγματικής παραμέτρου.

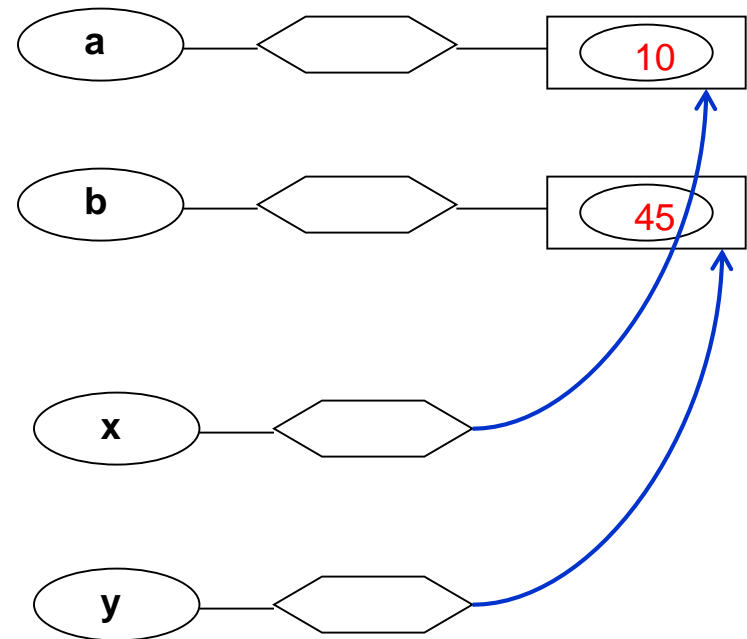
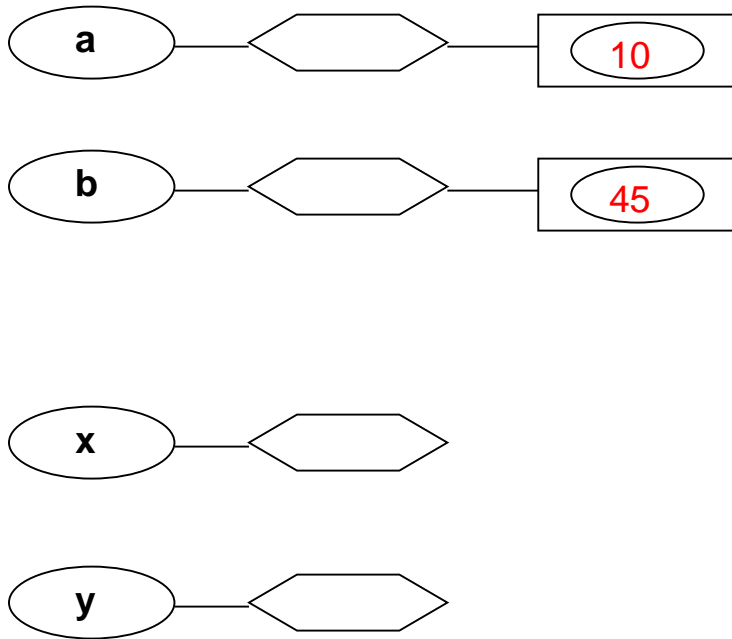


Κλήση με Αναφορά (2)

Σχηματικά:

Procedure P(VAR x, y)

Κλήση: P(a, b)



Κλήση με Τιμή - Αποτέλεσμα

- Όταν η πραγματική παράμετρος είναι μεταβλητή, αποαναφοροποιείται και η τιμή αντιγράφεται σε μια νέα θέση μνήμης, όπως στην Κλήση με Τιμή. Η θέση αυτή μνήμης, χρησιμοποιείται στο σώμα της υπορουτίνας. Κατά την έξοδο, η τιμή της τυπικής παραμέτρου αντιγράφεται στη θέση μνήμης της πραγματικής μεταβλητής.
- **ALGOL-W**



Κλήση με Όνομα

- Αφήνει τις πραγματικές παραμέτρους χωρίς να υπολογιστεί η τιμή τους, μέχρι το χρονικό σημείο χρήσης τους στην υπορουτίνα.
- Δηλαδή, οι πραγματικές παράμετροι αντιμετωπίζονται οι ίδιες σαν υπορουτίνες χωρίς παραμέτρους (thunk), που εκτελούνται και υπολογίζεται η τιμή τους (για να δοθεί στην τυπική παράμετρο), με το τρέχον ΠΑ του προγράμματος ή υπορουτίνας το οποίο καλεί την τρέχουσα υπορουτίνα.
- Τότε, η τυπική παράμετρος συνδέεται με την πραγματική παράμετρο, όπως στην Κλήση με Αναφορά.
- Ο υπολογισμός της πραγματικής παραμέτρου, γίνεται εξ αρχής, κάθε φορά που χρησιμοποιείται η αντίστοιχη τυπική παράμετρος.



- **ALGOL-60, SIMULA** (επιλογή του χρήστη)

Παράδειγμα

MAIN

```
VAR M: integer;  
    C: array [1..10] of integer;  
Procedure R (VAR i, j: integer)  
begin  
    i = i + 1;  
    j = j + 1;  
    write (i, j);  
end;
```

BEGIN

```
M:= 2;  
R (M, C[M]);
```

END.

Ερώτημα 1:

Με την εντολή `write(i,j)` τι θα τυπωθεί, αν έχουμε **Κλήση με Αναφορά;**

Απάντηση 1:

3, C [2] + 1

Ερώτημα 2:

Με την εντολή `write(i,j)` τι θα τυπωθεί, αν έχουμε **Κλήση με Όνομα;**

Απάντηση 2:

3, C [3] + 1

Ερώτημα 3:

Τι τιμή έχει το `C[3]` στο τέλος με **Κλήση με Όνομα;**

Απάντηση 3:

C [3]_{αρχικό} + 1



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- 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.



Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**



Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Γιάννης Γαροφαλάκης, 2015. «Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών. Υπορουτίνες». Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1091/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. 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».



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

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

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

