



Προγραμματισμός Υπολογιστών & Υπολογιστική Φυσική

Ενότητα 4: Δομές Ελέγχου

Νικόλαος Στεργιούλας
Τμήμα Φυσικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

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 παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.



ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ

Μέρος 4ο

ΝΙΚΟΛΑΟΣ ΣΤΕΡΓΙΟΥΛΑΣ



ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΟΙ ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΗΣ

Με τους τελεστές σύγκρισης, συγκρίνουμε τις τιμές δύο μεταβλητών ή δύο εκφράσεων.

$(a > b)$	(μεγαλύτερο)
$(a < b)$	(μικρότερο)
$(a >= b)$	(μεγαλύτερο ή ίσο)
$(a <= b)$	(μικρότερο ή ίσο)
$(a != b)$	(όχι ίσο)
$(a == b)$	(ίσο)

ΠΡΟΣΟΧΗ: συγκρίνουμε την ισότητα με το διπλό `==` και όχι με το απλό `=`.

ΑΛΗΘΗΣ/ΨΕΥΔΗΣ ΕΚΦΡΑΣΗ

Όταν μια έκφραση στην οποία χρησιμοποιούμε τελεστή σύγκρισης είναι **ΑΛΗΘΗΣ**, τότε η τιμή της είναι **1**.

π.χ. αν $i = 9$; τότε η $(i > 5)$ έχει τιμή 1.

Όταν μια έκφραση στην οποία χρησιμοποιούμε τελεστή σύγκρισης είναι **ΨΕΥΔΗΣ**, τότε η τιμή της είναι **0**.

π.χ. αν $i = 9$; τότε η $(i > 15)$ έχει τιμή 0.

Επίσης, κάθε έκφραση με τιμή διαφορετική από 0 χαρακτηρίζεται αληθής, ενώ κάθε έκφραση με τιμή 0 χαρακτηρίζεται ψευδής.

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    int a = 3;

    printf("%d\n", (a>3) );
    printf("%d\n", (a<3) );
    printf("%d\n", (a>=3) );
    printf("%d\n", (a<=3) );
    printf("%d\n", (a==3) );
    printf("%d\n", (a!=3) );

    return 0;
}
```

```
0
0
1
1
1
0
```

Η έκφραση

a += b;

είναι μια συντομογραφία της έκφρασης

a = a + b;

Ο ίδιος κανόνας ισχύει και για τους τελεστές -, *, /, % .

ΠΑΡΑΔΕΙΓΜΑ


```
#include <stdio.h>

int main(void)
{
    int a = 4, b = 2;

    a += 6;
    a *= b+3;
    a -= b+8;
    a /= b;
    a %= b+1;

    printf("Το αποτέλεσμα είναι = %d\n", a);

    return 0;
}
```



```
a = a + 6;
a = a * (b+3);
a = a - (b+8);
a = a / b;
a = a % (b+1);
```

Το αποτέλεσμα είναι = 2

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ

Οι συνήθεις λογικοί τελεστές γράφονται ως:

AND : &&

OR : ||

Παραδείγματα:

Η έκφραση $(10 == 10) \ \&\& \ (5 > 3)$ είναι αληθής, διότι και οι δύο επιμέρους εκφράσεις είναι αληθείς.

Η έκφραση $(10 == 10) \ || \ (5 < 3)$ είναι αληθής, διότι έστω μία επιμέρους έκφραση είναι αληθής.

Ο ΤΕΛΕΣΤΗΣ `sizeof`

Υπολογίζει πόσα bytes δεσμεύονται στη μνήμη για κάθε μεταβλητή. Π.χ.

```
#include <stdio.h>

int main(void)
{
    printf("char = %d bytes\n", sizeof(char));
    printf("int = %d bytes\n", sizeof(int));
    printf("float = %d bytes\n", sizeof(float));
    printf("double = %d bytes\n", sizeof(double));

    return 0;
}
```

```
char = 1 bytes
int = 4 bytes
float = 4 bytes
double = 8 bytes
```

Η ΕΝΤΟΛΗ `if`

Η `if` είναι από τις βασικότερες δομές ελέγχου της ροής ενός προγράμματος. Στην πιο απλή μορφή, η σύνταξη είναι:

```
if ( συνθήκη )  
{  
  ...  
  /* ομάδα εντολών */  
  ...  
}
```

Αν `συνθήκη = ΑΛΗΘΗΣ`, τότε εκτελείται η ομάδα εντολών μέσα στα άγκιστρα.

Αν `συνθήκη = ΨΕΥΔΗΣ`, τότε δεν εκτελείται η ομάδα εντολών μέσα στα άγκιστρα.

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    int x = 3;

    if( x == 0 )
    {
        printf("Η μεταβλητή x είναι 0.\n");
    }

    if( x != 0 )
    {
        printf("Η μεταβλητή x δεν είναι 0.\n");
    }

    return 0;
}
```

Η μεταβλητή x δεν είναι 0.

Η ΕΝΤΟΛΗ if

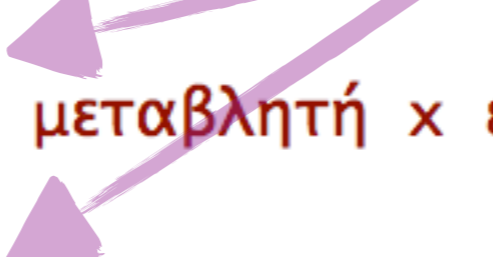
ΣΗΜΕΙΩΣΗ: Αν η ομάδα εντολών μέσα στα άγκιστρα είναι μόνο μία εντολή, τότε τα άγκιστρα είναι προαιρετικά, δηλ. αρκεί:

```
if( x == 0 )  
    printf("Η μεταβλητή x είναι 0.\n");  
  
if( x != 0 )  
    printf("Η μεταβλητή x δεν είναι 0.\n");
```

ΠΡΟΣΟΧΗ: Στο τέλος της συνθήκης του if δε βάζουμε ; !

(συντακτικό λάθος)

```
if( x == 0 );  
    printf("Η μεταβλητή x είναι 0.\n");  
  
if( x != 0 );  
    printf("Η μεταβλητή x δεν είναι 0.\n");
```



Η ΕΝΤΟΛΗ if

ΠΡΟΣΟΧΗ: Στη συνθήκη της εντολής if, **δεν πρέπει να συγχέουμε** τον τελεστή ισότητας **==** με τον τελεστή εκχώρησης **=**.

```
#include <stdio.h>

int main(void)
{
    int x = 3;

    if ( x == 3)
        printf("Η τιμή του x είναι %d\n", x);

    if ( x = 2) ← (άλλαξε η τιμή του x)
        printf("Η τιμή του x είναι %d\n", x);

    return 0;
}
```

```
Η τιμή του x είναι 3
Η τιμή του x είναι 2
```

Η ΕΝΤΟΛΗ if - else

Χρησιμοποιούμε το **else** όταν θέλουμε να εκτελεστεί μια συγκεκριμένη ομάδα εντολών στην περίπτωση που η συνθήκη ελέγχου στην if είναι ψευδής.

if (συνθήκη)

```
{  
...  
... /* ομάδα εντολών */  
...  
}
```

← (εκτελείται αν
συνθήκη = ΑΛΗΘΗΣ)

else

```
{  
...  
... /* ομάδα εντολών */  
...  
}
```

← (εκτελείται αν
συνθήκη = ΨΕΥΔΗΣ)

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    int x;

    printf("Δώσε έναν εκέριο αριθμό:\n");
    scanf("%d",&x);

    if (x > 0)
        printf("0 ακέριο αριθμός είναι θετικός.\n");
    else
        printf("0 ακέριο αριθμός είναι αρνητικός ή μηδέν.\n");

    return 0;
}
```

Δώσε έναν εκέριο αριθμό:

-4

0 ακέριο αριθμός είναι αρνητικός ή μηδέν.

ΠΑΡΑΔΕΙΓΜΑ: ΕΥΡΕΣΗ ΜΕΓΙΣΤΗΣ ΤΙΜΗΣ

```
#include <stdio.h>

int main(void)
{
    double price1, price2, price3, price4, max;

    printf("Δώσε τέσσερις τιμές: ");

    scanf("%lf%lf%lf%lf", &price1, &price2, &price3, &price4);

    if(price1 > price2)
        max = price1;
    else
        max = price2;

    if(price3 > max)
        max = price3;

    if(price4 > max)
        max = price4;

    printf("Η μέγιστη τιμή είναι %f\n", max);

    return 0;
}
```

ΣΥΝΕΧΟΜΕΝΕΣ ΕΝΤΟΛΕΣ `if - else`

Μια εντολή `if - else` μπορεί να **συνεχίζεται** με νέα εντολή `if - else`, π.χ.

```
if ( συνθήκη )
{
    ...
}
else if ( συνθήκη )
{
    ...
}
else if ( συνθήκη )
{
    ...
}
else
{
    ...
}
```

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    int x;

    printf("Δώσε έναν εκέραιο αριθμό:\n");
    scanf("%d",&x);

    if (x > 0)
        printf("0 ακέραιος αριθμός είναι θετικός.\n");
    else if (x < 0)
        printf("0 ακέραιος αριθμός είναι αρνητικός.\n");
    else
        printf("0 ακέραιος αριθμός είναι το μηδέν.\n");

    return 0;
}
```

Δώσε έναν εκέραιο αριθμό:

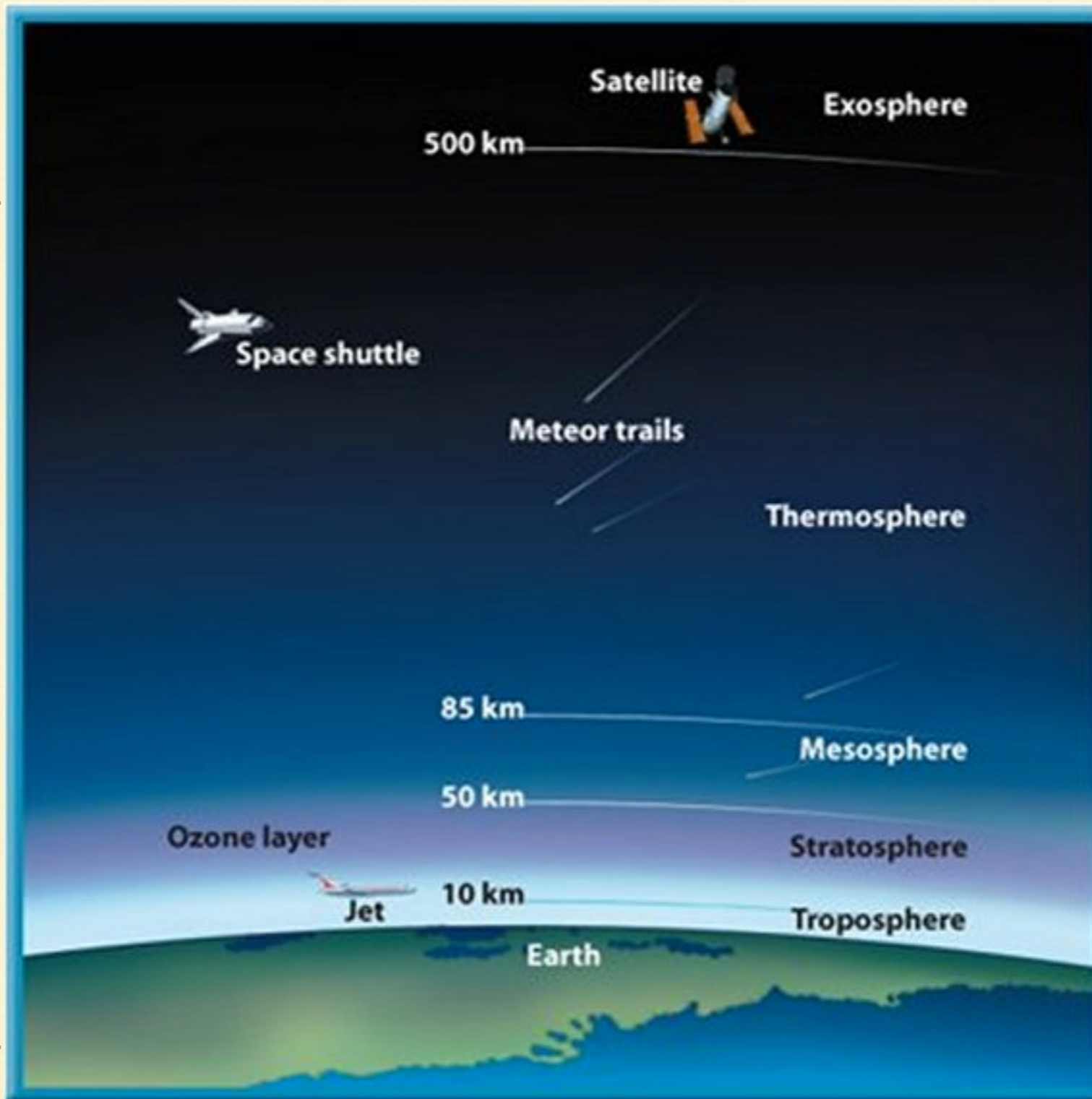
0

0 ακέραιος αριθμός είναι το μηδέν.

ΠΑΡΑΔΕΙΓΜΑ

Να γραφεί ένα πρόγραμμα που να υποδεικνύει την περιοχή της ατμόσφαιρας, με βάση το ύψος από το έδαφος.

<http://kudzucres.com/wwow/lessons/climate/atmospheric.html>



ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    float height;

    printf("Δώσε το ύψος σε km: ");

    scanf("%f", &height);

    if(height == 0.0)
        printf("Έδαφος\n");

    else if(height > 0.0 && height < 10.0)
        printf("Τροπόσφαιρα\n");

    else if(height >= 10.0 && height < 50.0)
        printf("Στρατόσφαιρα\n");

    else if(height >= 50.0 && height < 85.0)
        printf("Μεσόσφαιρα\n");

    else if(height >= 85.0 && height < 500.0)
        printf("Θερμόσφαιρα\n");

    else if(height >= 500.0)
        printf("Εξώσφαιρα\n");

    else
        printf("Το ύψος πρέπει να είναι θετικός αριθμός\n");

    return 0;
}
```

ΕΝΘΕΤΕΣ ΕΝΤΟΛΕΣ `if - else`

Γενικότερα, μια εντολή `if - else` μπορεί να περιέχει μέσα της μια **ένθετη** εντολή `if - else`, π.χ.

```
if ( συνθήκη )  
{
```

```
    if ( συνθήκη )  
    {  
        ...  
        ...  
    }  
    else {  
        ...  
        ...  
    }
```

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

← (ένθετη εντολή
`if - else`)

ΕΝΘΕΤΕΣ ΕΝΤΟΛΕΣ `if - else`

Γενικότερα, μια εντολή `if - else` μπορεί να περιέχει μέσα της μια **ένθετη** εντολή `if - else`, π.χ.

```
if ( συνθήκη )  
{  
    ...  
}  
else {
```

```
    if ( συνθήκη )  
    {  
        ...  
        ...  
    }  
    else {  
        ...  
        ...  
    }  
}
```

(ένθετη εντολή
`if - else`)



ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    int a = 10, b = 20, c = 30;

    if(a > 5)
    {
        if(b == 20)
            printf("Ναι, b=20.\n");

        if(c == 40)
            printf("Ναι, c=40.\n");
        else
            printf("'Οχι, c!=40.\n");
    }
    else
        printf("'Οχι, a<=5.\n");

    return 0;
}
```

Ναι, b=20.
'Οχι, c!=40.

Ο ΤΕΛΕΣΤΗΣ ?

Μια συντομογραφία της εντολής **if-else** είναι ο τελεστής **?** Αντί να γράψουμε

```
if ( συνθήκη )
    εντολή_1;
else
    εντολή_2;
```

μπορούμε να γράψουμε πιο σύντομα:

```
( συνθήκη ) ? εντολή_1 : εντολή_2;
```

Αν **συνθήκη = ΑΛΗΘΗΣ**, τότε **εκτελείται** η **εντολή_1**, αλλιώς, **εκτελείται** η **εντολή_2**.

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

int main(void)
{
    int a = 10, b = 20, c = 30;

    if(a > 5)
    {
        if(b == 20)
            printf("Ναι, b=20.\n");

        (c == 40) ? printf("Ναι, c=40.\n") : printf("'Οχι, c!=40.\n");
    }
    else
        printf("'Οχι, a<=5.\n");

    return 0;
}
```

Ναι, b=20.
'Οχι, c!=40.

Ο ΤΕΛΕΣΤΗΣ ?

Το αποτέλεσμα μιας εφαρμογής του τελεστή ? μπορεί να εκχωρηθεί άμεσα ως τιμή σε μια μεταβλητή. Π.χ. αντί να γράψουμε

```
if ( a > b )
    max = a;
else
    max = b;
```

μπορούμε να γράψουμε πιο σύντομα:

```
max = ( a > b ) ? a : b;
```

ΣΗΜΕΙΩΣΗ: Μπορούμε να έχουμε ένθετες εφαρμογές του τελεστή ? αλλά αυτό δε συνίσταται, καθώς το πρόγραμμα γίνεται πολύ δυσανάγνωστο.

Η ΕΝΤΟΛΗ `switch`

Είναι εναλλακτική κάποιων συνεχιζόμενων εντολών `if - else`, π.χ.

```
if ( έκφραση == τιμή_1 )
{
    εντολή_1;
}
else if ( έκφραση == τιμή_2 )
{
    εντολή_2;
}
else if ( έκφραση == τιμή_3 )
{
    εντολή_3;
}
else
{
    εντολή_4;
}
```



```
switch ( έκφραση )
{
    case τιμή_1:
        εντολή_1;
        break;

    case τιμή_2:
        εντολή_2;
        break;

    case τιμή_3:
        εντολή_3;
        break;

    default:
        εντολή_4;
        break;
}
```

Η ΕΝΤΟΛΗ `switch`

ΠΡΟΣΟΧΗ: Η εντολή `switch` δέχεται μόνο **ακέραια έκφραση** ως όρισμα, π.χ.

```
if ( a == 1 )
{
    εντολή_1;
}
else if ( a == 2 )
{
    εντολή_2;
}
else if ( a == 3 )
{
    εντολή_3;
}
else
{
    εντολή_4;
}
```



```
switch ( a )
{
    case 1:
        εντολή_1;
        break;

    case 2:
        εντολή_2;
        break;

    case 3:
        εντολή_3;
        break;

    default:
        εντολή_4;
        break;
}
```

Η ΕΝΤΟΛΗ `switch`

Ανάμεσα στις εντολές `case` και `break` δε χρειάζονται άγκιστρα.

Η `switch` ελέγχει διαδοχικά τις ακέραιες τιμές. Μόλις βρεθεί η έκφραση να είναι ίση με μία από τις ακέραιες τιμές, εκτελούνται οι εντολές αυτής της περίπτωσης και με το `break` το πρόγραμμα εξέρχεται από την εντολή `switch`.

Εάν η έκφραση δεν ισούται με καμία από τις ακέραιες τιμές, τότε εκτελούνται οι εντολές της περίπτωσης `default` και το πρόγραμμα εξέρχεται από την εντολή `switch`. Η περίπτωση `default` δεν είναι υποχρεωτική.

ΠΑΡΑΔΕΙΓΜΑ: ΑΠΛΟ CALCULATOR

```
#include <stdio.h>

int main()
{
    char operation;
    double a, b;

    printf("Δώσε την πράξη μεταξύ δύο αριθμών: ");
    scanf("%lf%c%lf", &a, &operation, &b);
    switch(operation)
    {
        case '+':
            printf("Άθροισμα = %f\n", a+b);
            break;

        case '-':
            printf("Διαφορά = %f\n", a-b);
            break;

        case '*':
            printf("Γινόμενο = %f\n", a*b);
            break;

        case '/':
            if(b != 0)
                printf("Πηλίκο = %f\n", a/b);
            else
                printf("0 διαιρέτης δε μπορεί να είναι μηδέν.\n");
            break;

        default:
            printf("Μη αποδεκτή πράξη\n");
            break;
    }

    return 0;
}
```

(δε θα χρειαστεί κενά, π.χ. δίνουμε 3.0+4.1 και όχι 3.0 + 4.1, διότι η ενδιάμεση μεταβλητή είναι char.)

ΠΑΡΑΔΕΙΓΜΑ: ΥΠΟΛΟΓΙΣΜΟΣ ΕΜΒΑΔΟΥ

```
#include <stdio.h>

#define PI 3.14159265358979323846

int main(void)
{
    int selection;
    double length, radius;

    printf("Επίλεξε 0 για τετράγωνο, 1 για κύκλο: ");
    scanf("%d", &selection);

    switch(selection)
    {
        case 0:
            printf("Δώσε το μήκος της πλευράς: ");
            scanf("%lf", &length);
            if(length <= 0.0)
            {
                printf("Μη-αποδεκτό μήκος.\n");
                return 0;
            }
            printf("Το εμβδιδόν του τετραγώνου είναι %f\n", length*length);
            break;

        case 1:
            printf("Δώσε την ακτίνα ");
            scanf("%lf", &radius);
            if(radius <= 0.0)
            {
                printf("Μη-αποδεκτή ακτίνα.\n");
                return 0;
            }
            printf("Το εμβιδό του κύκλου είναι %f\n", PI*radius*radius);
            break;

        default:
            printf("Μη-αποδεκτή επιλογή.\n");
            break;
    }
    return 0;
}
```

Αν οι εντολές που αντιστοιχούν σε δύο ή περισσότερες περιπτώσεις `case` είναι κοινές, μπορεί να γίνει συνένωση, π.χ.

```
switch ( a )
{
    case 1:
    case 2:
    case 3:
        εντολή_1;
    break;

    case 4:
    case 5:
        εντολή_2;
    break;

    default:
        εντολή_3;
    break;
}
```

Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, **Νικόλαος Στεργιούλας**
«Προγραμματισμός Υπολογιστών & Υπολογιστική Φυσική». Έκδοση: 1.0.
Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
http://opencourses.auth.gr/eclass_courses.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [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».



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

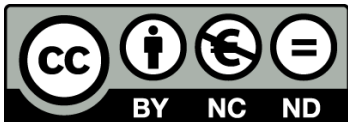
[1] <https://creativecommons.org/licenses/by-nc-nd/4.0/>





Τέλος ενότητας

Επεξεργασία: Νικόλαος Τρυφωνίδης
Θεσσαλονίκη, 20/09/2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

ΣΗΜΕΙΩΜΑΤΑ

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

