Ηλεκτρονικοί Υπολογιστές

Ενότητα 8: Επαναληπτικές Εντολές στη C++

Ζαχαρούλα Ανδρεοπούλου

Τμήμα Δασολογίας & Φυσικού Περιβάλλοντος

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

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

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.

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΧΤΑ
ΑΚΑΔΗΜΑΪΚΑ
ΜΑΘΗΜΑΤΑ

Επαναληπτικές Εντολές στη C++

Περιεχόμενα ενότητας 1/3

1. Επαναληπτικές εντολές
   i. Επαναληπτικές εντολές
   ii. Βρόχοι
   iii. Ο βρόχος for
   iv. Αλγόριθμος εκτέλεσης της for
   v. Παράδειγμα με for
   vi. Μετρητές
   vii. Παράδειγμα με μετρητές
   viii. Εντολές for που δεν εκτελούνται ούτε μία φορά
   ix. For με πολλές μεταβλητές και μία συνθήκη
   x. Πολλαπλά for

Περιεχόμενα ενότητας 2/3

xi. Εντολές τύπου while

xii. Εντολή while

xiii. Αλγόριθμος εκτέλεσης while

xiv. Nested while

xv. Βρόχος do while

xvi. Αλγόριθμος εκτέλεσης της do while

xvii. Διαφορά μεταξύ while και do while

xviii. Ατέρμονες βρόχοι

xix. Εντολή SWITCH

xx. Εντολή BREAK

xxi. Εντολή CONTINUE

Περιεχόμενα ενότητας 3/3

2. Ασκήσεις
   i. Άσκηση 1
   ii. Άσκηση 2
   iii. Άσκηση 3
   iv. Άσκηση 4
   v. Άσκηση 5
   vi. Άσκηση 6
   vii. Άσκηση 7

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Επαναληπτικές εντολές

Επαναληπτικές Εντολές στη C++

Επαναληπτικές εντολές

- Στον προγραμματισμό θέλουμε μια ή περισσότερες εντολές να εκτελούνται όχι 1 αλλά πολλές φορές.

- Για το λόγο αυτό υπάρχουν οι επαναληπτικές εντολές:
  - For loops βρόχος for
  - While loops βρόχος while
  - Do while βρόχος do while

- Για **συγκεκριμένο αριθμό επαναλήψεων** χρησιμοποιούμε το βρόχο **for**.

Βρόχοι

- Οι μηχανισμοί βρόχου στη C++ είναι παρόμοιοι με τους αντίστοιχους μηχανισμούς σε άλλες γλώσσες υψηλού επιπέδου.
- Οι τρεις εντολές βρόχου της C++ είναι η εντολή for, η εντολή while και η εντολή do-while.
- Ο κώδικας που επαναλαμβάνεται σε ένα βρόχο ονομάζεται **σώμα βρόχου (loop body).**
- Η διαδικασία της εκτέλεσης του σώματος βρόχου ονομάζεται **επανάληψη (iteration)** του βρόχου.

Ο βρόχος for

Ο βρόχος for χρησιμοποιείται όταν θέλουμε να έχουμε συγκεκριμένο αριθμό επαναλήψεων της εκτέλεσης μίας ή περισσότερων εντολών. Μεταβλητή ελέγχου της for είναι η i.

- For (i=1; i<=n; i+=1)

Αρχική τιμή μεταβλητής ελέγχου

Τελική τιμή (συνθήκη)

Ρυθμός μεταβολής

{…….

…… (περιοχή/πεδίο ελέγχου της for)

……. }

Για όσο ισχύει η συνθήκη εκτελούμε τις εντολές που ακολουθούν μέσα σε άγκιστρα.

Αλγόριθμος εκτέλεσης της for

- **Βήμα 1**
  Πρώτα παίρνει τιμή η μεταβλητή της FOR.
- **Βήμα 2**
  Μετά ελέγχεται η συνθήκη που ακολουθεί.
  - Αν η συνθήκη **δεν αληθεύει**, σταματά η εκτέλεση της FOR και το πρόγραμμα συνεχίζει από την εντολή μετά το πεδίο έλεγχου της FOR.
  - Αν η συνθήκη αληθεύει, εκτελούνται οι εντολές του πεδίου ελέγχου της FOR, μεταβάλλεται ο μετρητής, επιστρέφουμε στο Βήμα 2.

  - Int I;

    FOR (I=1; I<=5; I=I+1)

    {printf(/n/n “τυπωσε το %d”, I)}

Παράδειγμα με for

Παράδειγμα

```
For (i=1; i<=5; i=i+1)
{
cin >> a;
cout << endl;
cout << a;
}
```

- Οι εντολές cin και cout θα εκτελεστούν τόσες φορές όσο είναι να πάμε από το 1 (i=1) στο 5 (i<=5) με βήμα 1 (i=i+1). Το i είναι αυτό που μετράει τα βήματα και η συνθήκη είναι αυτή που καθορίζει μέχρι πότε η for θα εκτελείται. Ο μετρητής i (i=i+1), κάθε φορά που οι cin και cout εκτελούνται, αυξάνει κατά 1. Η for αυτή, θα εκτελείται όσο το I παραμένει μικρότερο ή γίνει ίσο με το 5.

Μετρητές

- i=i+1;
- i++; ⇒ Η αύξηση κατά 1 θα γίνει αφού εκτελεστεί
- ++i; ⇒ Η αύξηση κατά 1 θα γίνει πριν εκτελεστεί
- i--; ⇒ Η μείωση κατά 1 θα γίνει αφού εκτελεστεί
- -- i; ⇒ Η μείωση κατά 1 θα γίνει πριν εκτελεστεί
- a=a+b; ⇒ Η τιμή του a αυξάνει κατά b (βήμα/ρυθμός αύξησης b)
- a+=b; ⇒ Η τιμή του a αυξάνει κατά b (βήμα/ρυθμός αύξησης b)
- a-=b; ⇒ Η τιμή του a μειώνεται κατά b (βήμα/ρυθμός μείωσης b)

Παράδειγμα με μετρητές 1/2

Παράδειγμα

```
#include <iostream.h>
void main ( )
{
int i=0, x=0, y=0, j=0, a=2, z;
x=a*i++; cout<<“x = “<<endl;
y=a*++j; cout<<“y = “<<y;
cout<<endl<<“i = “<<endl;
cout<<“j = “<<j<<endl;
x=0;
a=1;
x+=4;
z=a+x; cout<<endl<<“z = “<<z;
y=0; y-=2; cout<<endl<<“y = “<<y;
x=0; +x=2; cout<<endl<<“x = “<<x; }
```

Παράδειγμα με μετρητές 2/2

Αποτελέσματα

x = 0
y = 2
i = 1
j = 1
z = 5
y = -2
x = 2

Εντολές for που δεν εκτελούνται ούτε μία φορά

- Επειδή η συνθήκη ελέγχεται πιν αρχίσει η εκτέλεση των εντολών που ανληκουν στη for, ενδέχεται οι εντολές αυτές να μην εκτελεστούν ούτε μία φορά.
- **Παράδειγμα**

```
for (i=b; i<c; i++)
    cout << a;
```

Αν b είναι μεγαλύτερο ή ίσον του c τότε η cout<<a; Δεν θα εκτελεστεί.

For με πολλές μεταβλητές και μία συνθήκη

- Σε μια for μπορούμε να έχουμε **περισσότερες μεταβλητές** με αρχικές τιμές και αντίστοιχες εντολές που μεταβάλλουν τις τιμές τους

  αλλά έχουμε πάντοτε **ΜΙΑ ΜΟΝΟ ΣΥΝΘΗΚΗ.**

- int I,j,k;

  for(i=0, j=2; j+1<=22; i=i+1, j=j+1)

  {k=i+j;

  printf(/n/n K=%d, K);}

Πολλαπλά for

- Μπορούμε να έχουμε εντολές for μέσα σε for αρκεί οι εξωτερικές να επικαλύπτουν **πλήρως** τις εσωτερικές.
- For (I=0; I<=n; I++)

```
{……..;
……..;
        for (j=0;  j<=n; j++)
        {……..;
        ……..;     Εσωτερική for
        ……..;}
……..;
……..;}
```

Εντολές τύπου while

- Απλή while
  - WHILE (συνθήκη) ……;

- Σύνθετη while
  - WHILE (συνθήκη)

    {……;

    ……; Πεδίο ελέγχου της while

    ……;} ΠΡΟΣΟΧΗ **χωρίς ;**

Εντολή while

- WHILE (συνθήκη)

  {…….;

  …….; Πεδίο ελέγχου της while

  ……;}

- Στην περίπτωση αυτή εκτελούνται συνεχώς οι εντολές που ανήκουν στο πεδίο ελέγχου της while, όσο ισχύει η συνθήκη **αλλά** η συνθήκη μεταβάλλεται και αυτή μέσα στο πεδίο ελέγχου.
  - int I=0; int B=10;

    While (I<b)

    { I=I+3; printf I}

Αλγόριθμος εκτέλεσης while

- **Βήμα 1**: ελέγχεται η συνθήκη
  - Αν η συνθήκη **δεν αληθεύει**, τότε η εκτέλεση συνεχίζεται από την πρώτη εντολή μετά το πεδίο ελέγχου της WHILE
  - Αν η συνθήκη **αληθεύει,** εκτελούνται οι εντολές στο πεδίο ελέγχου της While

- **Βήμα 2**: πάμε στο Βήμα 1. Στην περίπτωση αυτή εκτελούνται συνεχώς οι εντολές που ανήκουν στο πεδίο ελέγχου της while, όσο ισχύει η συνθήκη.
  - Στις for η έξοδος επιτυγχάνεται όταν τελειώσουν οι επαναλήψεις, ενώ στις while μπορούμε να βγούμε μόλις πάψει να ισχύει η συνθήκη.

Nested while

- Μπορούμε να έχουμε μια while μέσα σε άλλη, αρκεί η μια να καλύπτει πλήρως την άλλη.
- **While (συνθήκη-1)**

  {…….;

  …….;

  While (συνθήκη-2)

  {…….;

  …….;} εσωτερική while

  …….;

  …….;}

Βρόχος do while

- Η συνθήκη εξετάζεται **στο τέλος του πεδίου ελέγχου** του βρόγχου. **ΟΠΟΤΕ**, οι εντολές του πεδίου εκτελούνται τουλάχιστον **ΜΙΑ** φορά.
- Do

  {……;

  ……; Πεδίο έλεγχου της do while

  ……;}

  While (συνθήκη); **ΠΡΟΣΟΧΗ χρειάζεται ;**

Αλγόριθμος εκτέλεσης της do while

- **Βήμα 1**: εκτελούνται οι εντολές του πεδίου do while.

- **Βήμα 2**: εξετάζεται η συνθήκη.
  - Αν **ΔΕΝ ΑΛΗΘΕΥΕΙ** σταματά η εκτέλεση της DO WHILE και συνεχίζουμε με την εντολή μετά την WHILE (συνθήκη);
  - Αν **ΑΛΗΘΕΥΕΙ**, πηγαίνουμε στο Βήμα 1.

Παράδειγμα

```
DO
{printf( \n “diavase to  x”);
Scanf (%d, &x);
S=s+x;
I=I+1;}
While  (I<=n);
```

Θα διαβάσει τουλάχιστον ένα χ.

Διαφορά μεταξύ while και do while

- Η σημαντική διαφορά μεταξύ των βρόχων while και do while είναι πότε ελέγχεται η λογική έκφραση ελέγχου.
- Σε μια εντολή while, η λογική έκφραση ελέγχεται πριν από την εκτέλεση του σώματος του βρόχου.
- Σε μια εντολή do while, εκτελείται πρώτα το σώμα του βρόχου και η λογική έκφραση ελέγχεται μετά την εκτέλεση του σώματος βρόχου. Συνεπώς, η εντολή d while εκτελεί πάντα το σώμα του βρόχου τουλάχιστον μία φορά.

Ατέρμονες βρόχοι

- Ένας βρόχος while, do while ή for δεν τερματίζεται όσο η λογική έκφραση ελέγχου είναι αληθής.

- Η λογική έκφραση περιέχει μία μεταβλητή, η οποία θα μεταβληθεί από το σώμα βρόχου και συνήθως η τιμή αυτής της μεταβλητής αλλάζει με τέτοιο τρόπο που τελικά καθιστά τη λογική έκφραση ψευδή και συνεπώς τερματίζει το βρόχο.

- Εάν όμως κάνουμε λάθος και γράψουμε το πρόγραμμα έτσι ώστε η λογική έκφραση να είναι πάντα αληθής, τότε ο βρόχος θα τρέχει για πάντα. Ένας βρόχος που τρέχει για πάντα λέγεται **ατέρμων βρόχος (infinite loop)**.

Εντολή SWITCH 1/2

- Η εντολή switch χρησιμοποιεί πολλαπλές διακλαδώσεις.

- Όταν εκτελείται μια εντολή switch εκτελείται μία διακλάδωση από ένα σύνολο διαφορετικών διακλαδώσεων. Η επιλογή της διακλάδωσης που θα εκτελεστεί καθορίζεται από μια έκφραση ελέγχου (controlling expression), η οποία γράφεται μέσα σε παρενθέσεις μετά τη δεσμευμένη λέξη switch.

- Όταν εκτελείται η εντολή switch, η έκφραση ελέγχου αποτιμάται και ο υπολογιστής εξετάζει τις τιμές των σταθερών που βρίσκονται μετά από κάθε πρόταση case.

Εντολή SWITCH 2/2

- Αν βρει μια τιμή που είναι ίση με την τιμή της έκφρασης ελέγχου, εκτελεί τον κώδικα για αυτήν την πρόταση case. Δεν μπορούμε να έχουμε δύο προτάσεις case που να ακολουθούνται από την ίδια τιμή της σταθεράς επειδή θα δημιουργηθεί μία απροσδιόριστη εντολή.

- Η εντολή switch τερματίζει όταν εμφανιστεί μια εντολή break ή όταν το πρόγραμμα φτάσει στο τέλος της εντολής switch.

Εντολή BREAK

- Η εντολή break αποτελείται από την δεσμευμένη εντολή break και ένα ελληνικό ερωτηματικό.
- Η εντολή break χρησιμοποιείται για την διακοπή της εκτέλεσης μιας επαναληπτικής εντολής (**while, for, do …. while**).
- Η εκτέλεση του προγράμματος μεταφέρεται στην εντολή που υπάρχει αμέσως μετά την while, for, do …. while.

Εντολή CONTINUE

- Η εντολή continue αποτελείται από την δεσμευμένη εντολή continue και ένα ελληνικό ερωτηματικό.

- Όταν εκτελείται η εντολή continue τερματίζει την τρέχουσα επανάληψη του σώματος βρόχου της πλησιέστερης περικλείουσας εντολής βρόχου.

- **ΠΡΟΣΟΧΗ**
  Η εντολή continue μέσα σε ένα βρόχο for μεταφέρει τον έλεγχο στην ενημερωμένη έκφραση. Συνεπώς, κάθε μεταβλητή ελέγχου βρόχου θα ανανεώνεται αυτόματα μετά από την εκτέλεση της εντολής continue .

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Ασκήσεις

Επαναληπτικές Εντολές στη C++

Άσκηση 1

Να τυπώσετε άρτιους αριθμούς 2,….,N.

```
Int  I, N;
For (I=2; I<=N; I=I+2)
{
Printf(/n/n %d, I)}
```

Κάποιες εντολές for (πεδία ελέγχου) δεν εκτελούνται ποτέ γιατί δεν μπορούμε να πάμε από την αρχική τιμή της μεταβλητής ελέγχου στην τελική τιμή.

- for (I=1; I<=50; I--)
  for (j=10; j>=0; j++)

Άσκηση 2 1/2

Να συνταχθεί πρόγραμμα που να υπολογίζει το άθροισμα Σ = 1+2+3+4+5+……+n.

```
/* Άθροισμα Σ = 1+2+3+4+5+……+n */
#include <iostream.h>
#include <stdio.h>
Void main ( )
{
Int i, s, n;
s=0; // midenismos athrismatos
Printf (“\n\n dose teliko n…”);
Scanf (“%d”, &n);
```

Άσκηση 3 2/2

```
For (i=1; i<=n; i=i+1)
{
s=s+i;
}

Printf (“\n\n to athroisma einai %d”,s);
}
Exit(0);
```

Άσκηση 4 1/2

Να συνταχθεί πρόγραμμα που να υπολογίζει το άθροισμα Σ = 2+4+6+8+……+n.

```
/* Άθροισμα Σ = 2+4+6+8+……+n */
#include <iostream.h>
#include <stdio.h>
Void main ( )
{
Int i, s, n;
s=0; // midenismos athrismatos
Printf (“\n\n dose teliko n…”);
```

Άσκηση 4 2/2

```
Scanf (“%d”, &n);
For (i=2; i<=n; i=i+2)
{
s=s+i;
}

Printf (“\n\n to athroisma einai %d”,s);
}
Exit(0);
```

Άσκηση 5 1/2

Να συνταχθεί πρόγραμμα που να υπολογίζει το άθροισμα $\Sigma = 1^2+3^2+5^2+\ldots\ldots\ldots+n^2$.

```
/* Άθροισμα Σ = 1^2+3^2+5^2+.....+n^2 */
#include <iostream.h>
#include <stdio.h>
#include <math.h>
Void main ( )
{
Int i, s, n;
s=0; // midenismos athrismatos
Printf ("\n\n dose teliko n...");
```

Άσκηση 5 2/2

```
Scanf (“%d”, &n);
For (i=1; i<=n; i=i+2)
{
s=s+pow(i,2);
}
Printf (“\n\n to athroisma einai %d”,s);
}
Exit(0);
```

Άσκηση 6 1/2

Να συνταχθεί πρόγραμμα που να υπολογίζει το άθροισμα $\Sigma = 2^3+4^3+6^3+8^3+\ldots\ldots\ldots+n^3$.

```
/* Άθροισμα Σ = 2^3+4^3+6^3+8^3+.....+n^3 */
#include <iostream.h>
#include <stdio.h>
#include <math.h>
Void main ( )
{
Int i, s, n;
s=0; // midenismos athrismatos
Printf (“\n\n dose teliko n…”);
```

Άσκηση 6 2/2

```
Scanf (“%d”, &n);
For (i=2; i<=n; i=i+2)
{
s=s+pow( i , 3);
}
Printf (“\n\n to athroisma einai %d”,s);
}
Exit(0);
```

Άσκηση 7

Να συνταχθεί πρόγραμμα που να εκτυπώνει άρτιους αριθμούς 2,…,n. Να τυπώνεται: 1os ARTIOS = 2

2os ARTIOS = 4

3os ARTIOS = 6

……..

```
#  include <stdio.h>
void main ( )
{ Int num, n, j;
j=0; //Μηδενίζω το μετρητή
printf (“\n\n dwse to euros emfanisis artiwn:”);
scanf (“%d”, &n);//διαβάζει το n
        for (num=2; num<=n; num+=2)
        { j++;
        printf(“\n\n %d os ARTIOS = %d”,  j, num);}}
exit(0) ;
```

Βιβλιογραφία

- Jamsa, K. 1999. Εισαγωγή στη… C++. Μετάφραση: Τ. Άλβας. Εκδόσεις Κλειδάριθμος. Αθήνα
- Λάζος, Κ. 2004. C++: Θεωρία και πράξη. 2η Έκδοση. Θεσσαλονίκη
- Savitch, W. 2013. Πλήρης C++. 4η έκδοση. Μετάφραση: Σ. Κατσαβούνης. Εκδόσεις Τζιόλα. Αθήνα

- Ανδρεοπούλου, Ζ. 2011. Ηλεκτρονικοί Υπολογιστές. Πηγή στο Διαδίκτυο: http://www.for.auth.gr/uploads/pages/HLEKTRONIKOI_YPOLOGISTES_2014_theory_and_practise.pdf

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Τέλος ενότητας

Επεξεργασία: <Χριστιάνα Κολιούσκα>
Θεσσαλονίκη, <Χειμερινό εξάμηνο 2014-2015>

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημειώματα

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.