



Εισαγωγή στην πληροφορική

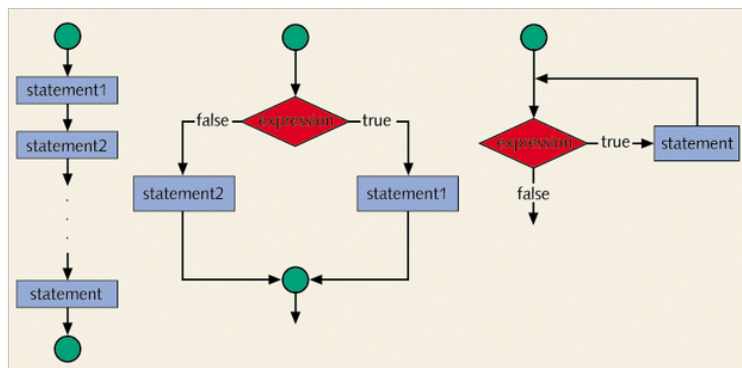
Βασίλειος Βεσκούκης
Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

*Η γλώσσα προγραμματισμού C++
Επαναληπτικές δομές*

Δομές ελέγχου 2: επαναληπτική εκτέλεση

Δομημένος προγραμματισμός

- Ακολουθία ($s_1 ; s_2$)
- Επιλογή (`if c then s_1 else s_2`)
- Επανάληψη (`while c do s`)



Δομές ελέγχου 2: επαναληπτική εκτέλεση

Επαναληπτική χαρακτηρίζεται η εκτέλεση ενός συνόλου εντολών ανάλογα με την ισχύ, δηλαδή την **αλήθεια** ή το **ψεύδος** μιας **λογικής συνθήκης**

Για ποιους λόγους είναι χρήσιμη η επαναληπτική εκτέλεση;

- Γιατί να επαναλάβουμε τις **ίδιες πράξεις** σε **όλα τα μέλη ενός πληθυσμού** (πχ: υπολογισμός της απόστασης διαδοχικών σημείων)
- Γιατί να προσπαθήσουμε να βρούμε ένα αποτέλεσμα επαναλαμβάνοντας υπολογισμούς μέχρις ότου ισχύει μια συνθήκη (πχ: επίλυση εξίσωσης)
- Για να επαληθεύσουμε την εγκυρότητα κάποιων δεδομένων που δίνονται

Τρεις δομές επαναληπτικής εκτέλεσης στη C++

- **while** (συνθήκη)
 εντολή
- **do**
 εντολή
 while (συνθήκη)
- **for** (εντολή_αρχικοποίησης; **συνθήκη**; εντολή_ενημέρωσης)
 εντολή

Δρ. Βασίλειος Βεσκούκης

Επαναληπτική εκτέλεση

Η λογική συνθήκη καθορίζει

- αν θα εκτελεστεί η εντολή
- "πότε" θα σταματήσει να εκτελείται
- αν θα σταματήσει να εκτελείται

Ο αριθμός των επαναλήψεων (φορών που θα εκτελεστεί η εντολή) μπορεί να είναι

- Γνωστός εξαρχής (συνήθης χρήση δομής for())
- Αγνώστος και επιπλέον αδιάφορος, εφόσον κάποτε η επανάληψη ολοκληρώνεται

Συνήθη προβλήματα μαθητευομένων προγραμματιστών

- Λάθος ή καθόλου χρήση του block εντολών
- Λάθος στη λογική συνθήκη
- Κακή ή καθόλου στοίχιση προγράμματος ιδιαίτερα όταν υπάρχουν και άλλες δομές ελέγχου μέσα σε ένα block εντολών (αναγνωσιμότητα)

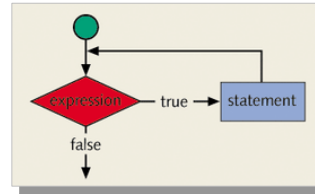
Δρ. Βασίλειος Βεσκούκης

Η δομή while

while (expression)
statement

Λειτουργία:

- Αν η **συνθήκη** είναι αληθής, εκτελείται η **εντολή** και η ισχύς της συνθήκης ελέγχεται και πάλι.
- Η διαδικασία έλεγχος αληθείας → εκτέλεση επαναλαμβάνεται μέχρις ότου να **μην ισχύει** η **συνθήκη**.



Παρατηρήσεις

- Αν κατά τη στιγμή της εισόδου η **συνθήκη** είναι ψευδής, η **εντολή** δεν θα εκτελεστεί **καμία φορά**
- Αν η **εντολή** δεν μεταβάλλει την τιμή της **συνθήκης** από αληθή σε ψευδή, τότε η **εντολή** θα εκτελείται επ' άπειρον...
- Κατά τον πρώτο έλεγχο ισχύος της **συνθήκης**, αυτή πρέπει να έχει αρχική τιμή.

Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Ευρεση του αθροίσματος 10 αριθμών

```
int numbers;           → Το πλήθος των αριθμών
int sum;               → Το άθροισμα των αριθμών
int datum;            → Κάθε αριθμός που διαβάζεται από το πληκτρολόγιο

numbers=0;
sum=0;                → Αρχικές τιμές

while (numbers<10)    → Όσο ισχύει η συνθήκη αυτή, θα εκτελείται η εντολή
{
  cout<<"Όσοε έναν αριθμό :";
  cin>>datum;         → Ανάγνωση ενός αριθμού
  sum=sum+datum;      → Πρόσθεση του αριθμού που διαβάστηκε στο sum
  numbers=numbers+1; → Αύξηση κατά 1 του μετρητή των αριθμών που έχουν αθροιστεί
}

cout<<"Το άθροισμα των 10 αριθμών είναι "<<sum;
```

Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Μια βελτιωμένη έκδοση του προηγούμενου προγράμματος

```
int how_many;
int numbers;
int sum;
int datum;

numbers=0;
sum=0;

cout<<"Πόσοι αριθμοί θα αθροιστούν;";
cin>>how_many;

while (numbers<how_many)
{
    cout<<"Ήώσε έναν αριθμό :";
    cin>>datum;

    sum=sum+datum;

    numbers=numbers+1;
}

cout<<"Το άθροισμα των"<<how_many<<" αριθμών είναι "<<sum;
```

Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Μια (ακόμη) βελτιωμένη έκδοση του προγράμματος

```
int how_many;
int numbers;
int sum;
int datum;

numbers=0;
sum=0;

cout<<"Πόσοι αριθμοί θα αθροιστούν;";
cin>>how_many;

while (numbers<how_many)
{
    cout<<"Ήώσε έναν αριθμό :";
    cin>>datum;

    sum=sum+datum;

    numbers=numbers+1;
}

cout<<"Το άθροισμα των"<<how_many<<" αριθμών είναι "<<sum;
```

Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Τι (μπορεί να) πέρασε απαραίτητο...

- Ο ορισμός των μεταβλητών που χρησιμοποιήθηκαν (συντακτικό σφάλμα)
- Η αρχικοποίηση με την τιμή μηδέν του αθροιστή και του μετρητή αριθμών
- Το block εντολών θα εκτελεστεί μόνο αν ισχύει η συνθήκη

Προς αποφυγή σφαλμάτων

- Στοιχίζουμε σωστά το πρόγραμμα
- Προσθέτουμε σχόλια
- Εκτελούμε το πρόγραμμα με δοκιμαστικά δεδομένα που δίνουν γνωστό σε μας αποτέλεσμα και συγκρίνουμε τα αποτελέσματα με τα αναμενόμενα

Δρ. Βασίλειος Βεσκούκης

Η δομή do ... while

do

statement

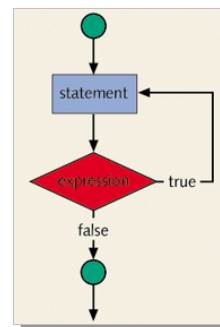
while (**expression**)

Λειτουργία:

- Η **εντολή** εκτελείται μία φορά και **κατόπιν** ελέγχεται η ισχύς της συνθήκης. Αν η **συνθήκη** είναι αληθής, εκτελείται και πάλι η **εντολή**.
- Η διαδικασία "εκτέλεση → έλεγχος συνθήκης" επαναλαμβάνεται μέχρις ότου να μην ισχύει η **συνθήκη**.

Παρατηρήσεις

- Η **εντολή** θα εκτελεστεί **τουλάχιστον μία φορά**
- Αν η **εντολή** δεν μεταβάλλει την τιμή της **συνθήκης** από αληθή σε ψευδή, τότε η **εντολή** θα εκτελείται επ' άπειρον...



Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Το ίδιο πρόγραμμα με χρήση do..while

```
int how_many, numbers, sum, datum;
numbers=1;
sum=0;
do
{
    cout<<"Πόσοι αριθμοί θα αθροιστούν; ";
    cin>>how_many;
}
while (how_many<=0);

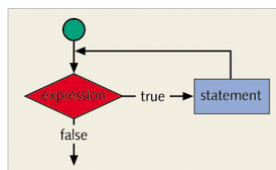
do
{
    cout<<"Ώστε τον αριθμό "<<numbers<<" από "<<how_many<<": ";
    cin>>datum;
    sum=sum+datum;
    numbers=numbers+1;
}
while (numbers<=how_many);
cout<<"\n\nΤο άθροισμα των "<<how_many<<" αριθμών είναι "<<sum<<"\n";
```

Εντολή { } Θα εκτελεστεί τουλάχιστον μία φορά

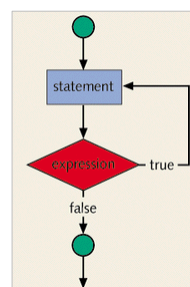
Δρ. Βασίλειος Βεσκούκης

Επαναληπτική εκτέλεση

Μια σύγκριση των δομών while... και do...while



```
int counter;
counter=100;
while (counter<10)
{
    cout<<"counter="<<counter<<"\n";
    counter++;
}
```



```
counter=100;
do
{
    cout<<"counter="<<counter<<"\n";
    counter++;
}
while (counter<10);
```

Δρ. Βασίλειος Βεσκούκης

Η δομή for ()

for (εντολή_αρχικοποίησης; συνθήκη; εντολή_ενημέρωσης)
εντολή

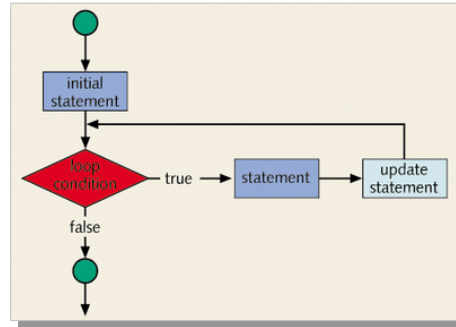
for (initial statement; loop condition; update statement)
statement

Λειτουργία

- Εκτελείται η εντολή_αρχικοποίησης
- Ελέγχεται η συνθήκη και αν ισχύει, εκτελείται η εντολή, διαφορετικά το for τερματίζει.
- Εκτελείται η εντολή_ενημέρωσης

Παρατηρήσεις

- Σύνθετη και ισχυρή εντολή, ικανή να χρησιμοποιείται ως μόνη εντολή επανάληψης
- Χρειάζεται προσοχή στην κατανόηση της σειράς εκτέλεσης των εντολών



Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Το προηγούμενο πρόγραμμα γραμμένο με for ()

```
int counter;
for (counter=1; counter<10; counter++)
{
    cout<<"counter="<<counter<<"\n";
}

counter=1;
while (counter<10)
{
    cout<<"counter="<<counter<<"\n";
    counter++;
}
```

Annotations for the for loop:

- Εντολή αρχικοποίησης (points to counter=1)
- Συνθήκη εκτέλεσης (points to counter<10)
- Εντολή ενημέρωσης (points to counter++)
- Εντολή που θέλουμε να επαναλαμβάνεται (points to the cout statement)

Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

Υπολογισμός μέσου όρου

```
int main()
{
    int how_many;
    int sum=0;
    int datum;
    int average;
    int counter;

    do
    {
        cout<<"\n"<<"Πόσα το πλήθος των αριθμών :";
        cin>>how_many;
    }
    while (how_many<=0);

    for (counter=1; counter<=how_many; counter++)
    {
        cout<<"Πόσα τον αριθμό "<<counter<<" :";
        cin>>datum;
        sum+=datum;
    }

    average=sum/how_many;
    cout<<"\n"<<"Μέσος όρος ="<<average<<"\n";
}
```

Δρ. Βασίλειος Βεσκούκης