



Προγραμματισμός Υπολογιστών

Δομές επανάληψης (2-3)

Νικόλαος Ζ. Ζάχαρης
Καθηγητής

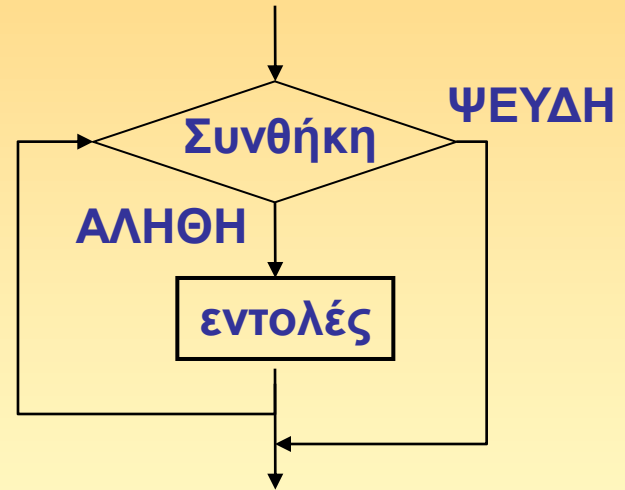
Πανεπιστήμιο Δυτικής Αττικής
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Η δομή επανάληψης while

Η δομή **while** όπως και η **for** μας επιτρέπει την επανάληψη των εντολών ενώ αληθεύει μία συνθήκη. Η σύνταξη της **while** είναι όπως παρακάτω :

```
while(Συνθήκη)  
{  
    εντολές;  
}
```

Όπως φαίνεται και στο διάγραμμα ροής η δήλωση **while** έχει μια **Συνθήκη** η εκτίμηση της οποίας αν είναι 0 (ΨΕΥΔΗΣ) οδηγεί στον τερματισμό της επανάληψης, διαφορετικά αν έχει τιμή διάφορη του 0 (ΑΛΗΘΗΣ) τότε εκτελούνται οι **εντολές**. Με το πέρας της εκτέλεσης των εντολών επαναλαμβάνεται ο έλεγχος της **Συνθήκης**.



Η δομή επανάληψης while

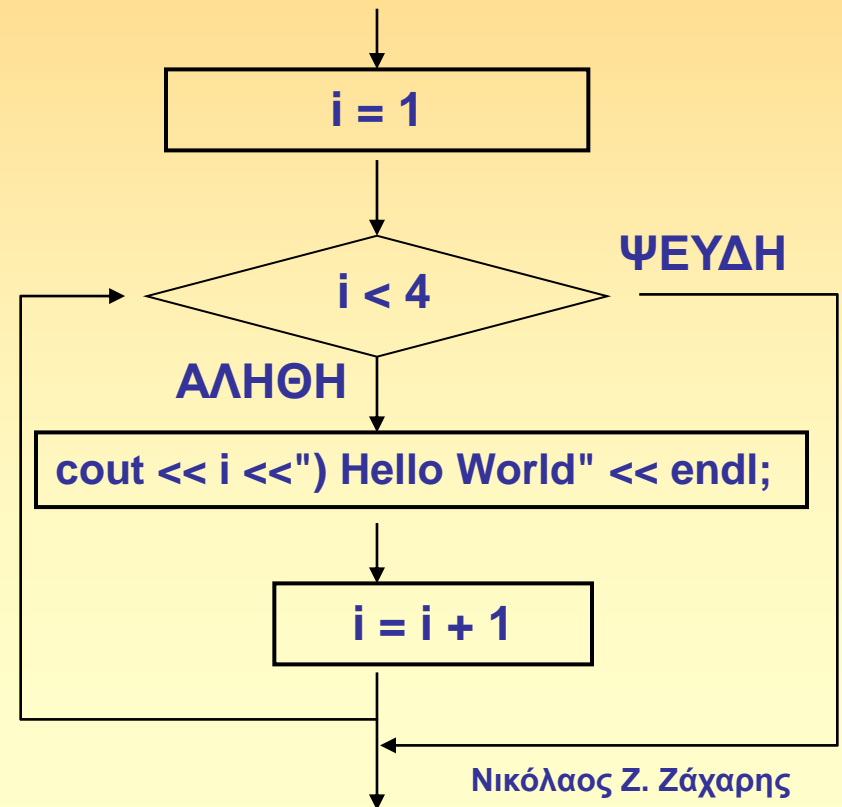
Η σύνταξη της while είναι παρόμοια με τη δομή if μόνο που στην if οι εντολές εκτελούνται μόνο μία φορά αν η συνθήκη είναι αληθής, ενώ στη while επαναλαμβάνεται η εκτέλεσή τους, εφόσον η συνθήκη παραμένει αληθής.

Η while έχει πιο απλή σύνταξη σε σχέση με τη for γιατί δεν χρειάζεται να δηλώσουμε ούτε αρχική τιμή ούτε το βήμα. Βέβαια και εδώ θα πρέπει να βρούμε έναν μηχανισμό ώστε να τερματίσουμε την επανάληψη.

```
int main(int argc, char *argv[])
{
    int i = 1;
    while(i < 4)
    {
        printf("%d) Hello World\n", i);
        i = i + 1;
    }

    return 0;
}
```

```
1) Hello World
2) Hello World
3) Hello World
```



Παράδειγμα

Χρησιμοποιώντας την δήλωση **while** δημιουργήστε ένα πρόγραμμα που διαβάζει τις θερμοκρασίες μιας εβδομάδας και εκτυπώνει στο τέλος την μέση θερμοκρασία.

Αρχή Αλγόριθμου

sum = 0

plithos = 1

Ενώ plithos <= 7 Επανάλαβε

 Διάβασε A

 plithos = plithos + 1

 sum = sum + A

Τέλος Επανάληψης

Εκτύπωσε sum / 7

Τέλος Αλγόριθμου

```
Type temperature for 1 day : 12
Type temperature for 2 day : 13
Type temperature for 3 day : 14
Type temperature for 4 day : 14
Type temperature for 5 day : 13
Type temperature for 6 day : 12
Type temperature for 7 day : 12
Average temperature : 12.857143
```

```
int main(int argc, char *argv[])
{
    int plithos = 1;
    float A, sum = 0.0;

    while(plithos <= 7)
    {
        printf(" Type temperature for %d day : ", plithos);
        scanf("%f", &A);
        sum = sum + A;
        plithos = plithos + 1;
    }
    printf("Average temperature : %f", sum / 7.0);

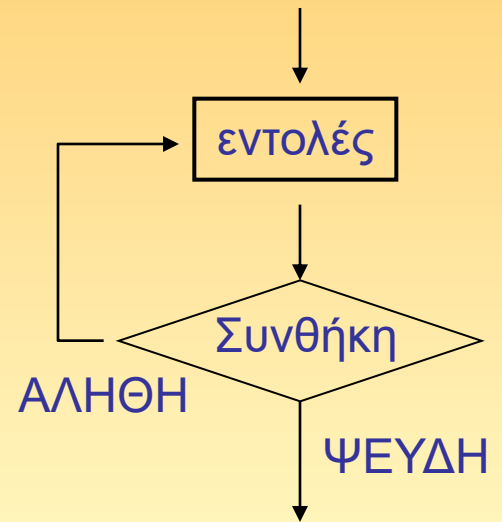
    return 0;
}
```

Η δήλωση do.. while

Η δήλωση **do..while** όπως και οι δηλώσεις **while** και **for** μας επιτρέπουν την επανάληψη των εντολών ενώ αληθεύει μία συνθήκη. Η σύνταξη της **do..while** είναι όπως παρακάτω :

```
do {  
    εντολές;  
} while(Συνθήκη);
```

Όπως φαίνεται και στο διάγραμμα ροής η δήλωση **do..while** σε αντίθεση με την **while** πρώτα εκτελεί τις **εντολές** μια φορά και εν συνεχεία κάνει εκτίμηση της **Συνθήκης**. Αν το αποτέλεσμα της συνθήκης είναι 0 (ΨΕΥΔΗΣ) τότε τερματίζεται και η εκτέλεση της δήλωσης. Διαφορετικά αν η συνθήκη έχει τιμή διάφορη του 0 (ΑΛΗΘΗΣ) τότε επαναλαμβάνεται και η εκτέλεση των **εντολών**.



Παράδειγμα

Διαβάστε ακεραίους αριθμούς συνέχεια από το πληκτρολόγιο μέχρι του σημείου να διαβάσετε έναν αρνητικό αρνητικό. Στο τέλος εκτυπώστε το πλήθος και το άθροισμα των αριθμών.

```
int main(int argc, char *argv[]) {
int a = 0, plithos = 0, sum = 0;
do
{
printf("Type a positive number : ");
scanf("%d", &a);
if(a >= 0) {
sum += a;
plithos += 1;
}
} while(a >= 0);

printf("Plithos = %d   Sum = %d\n", plithos, sum);
return 0;
}
```

Ενναλακτική προσέγγιση

```
int main(int argc, char *argv[]) {
int a = 0, plithos = 0, sum = 0;
do
{
printf("Type a positive number : ");
scanf("%d", &a);
sum += a;
plithos += 1;
} while(a >= 0);
printf("Plithos = %d   Sum = %d\n", plithos-1, sum - a);
return 0;
}
```