

Επαναληπτικές δομές

Η δήλωση for

Η σύνταξη της for

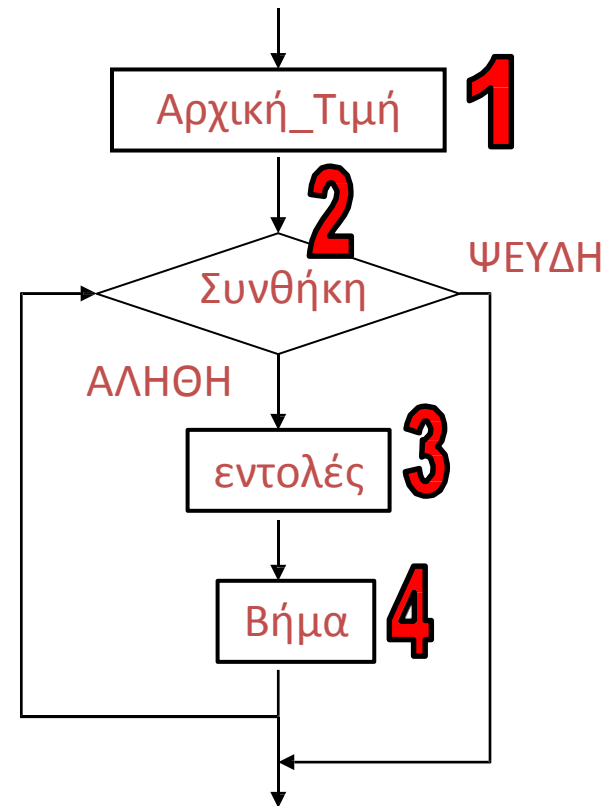
```
for(Αρχική_Τιμή; Συνθήκη; Βήμα) {  
    εντολές;  
}
```

1) Αρχικά θα εκτελεστεί για μια μόνο φορά η **Αρχική_Τιμή**. Είναι το σημείο που θα αποδίδουμε στο μετρητή της επανάληψης μια τιμή.

2) Θα γίνει σύγκριση της **Συνθηκης** και αν είναι ψευδής τερματίζει η επανάληψη, διαφορετικά

3) Θα εκτελεστούν οι **εντολές**

4) Θα εκτελεστεί το **Βήμα**. Είναι το σημείο που ο μετρητής αυξάνει ή μειώνει τιμή και συνεχίζει στο βήμα 2



Πόσες φορές θα εκτελεστούν οι εντολές;

```
int i;  
for(i=1456; i<13; i=i+2) {  
    εντολές;  
}
```

Διαφορετικοί τρόποι έκφρασης

Πρώτο παράδειγμα επανάληψης

```
int i;  
for(i = 1; i <= 3; i=i+1) {  
    εντολές;  
}
```

Δεύτερο παράδειγμα επανάληψης

```
int i;  
for(i = 50; i <= 150; i=i+50) {  
    εντολές;  
}
```

Τρίτο παράδειγμα επανάληψης

```
int i;  
for(i = 150; i >= 50; i=i-50) {  
    εντολές;  
}
```

Και στις τρεις περιπτώσεις οι εντολές θα εκτελεστούν 3 φορές.

Παράδειγμα for

Δημιουργείστε μια εφαρμογή στην οποία θα διαβάζεται τις τιμές δύο ακεραίων αριθμών , έστω A και B και θα υπολογίζεται το άθροισμα όλων των αριθμών ανάμεσα στα A και B.

Για παράδειγμα αν οι αριθμοί είναι $A = 5$ και $B = 8$ τότε η εφαρμογή μας θα πρέπει να υπολογίσει το άθροισμα όλων των αριθμών $5 + 6 + 7 + 8$ και να εκτυπώσει σαν αποτέλεσμα το 26.

Ο πηγαίος κώδικας

```
public class ForLoop {  
  
    public static void main(String[] args) {  
  
        System.out.print("Type first number ");  
        int first = new java.util.Scanner(System.in).nextInt();  
  
        System.out.print("Type second number ");  
        int second = new java.util.Scanner(System.in).nextInt();  
  
        int i, sum = 0;  
  
        for(i = first; i <= second; i++)  
        {  
            sum = sum + i;  
        }  
  
        System.out.println("Sum = " + sum);  
    }  
}
```

Τι τιμή έχει η μεταβλητή sum με το πέρας της επανάληψης;

```
int x = 0, y = 10, sum = 0;
```

```
for(x = 25; (x <= 30) && (y < 20); x++)
```

```
{
```

```
    sum = sum + 10;
```

```
    y = y + 5;
```

```
}
```

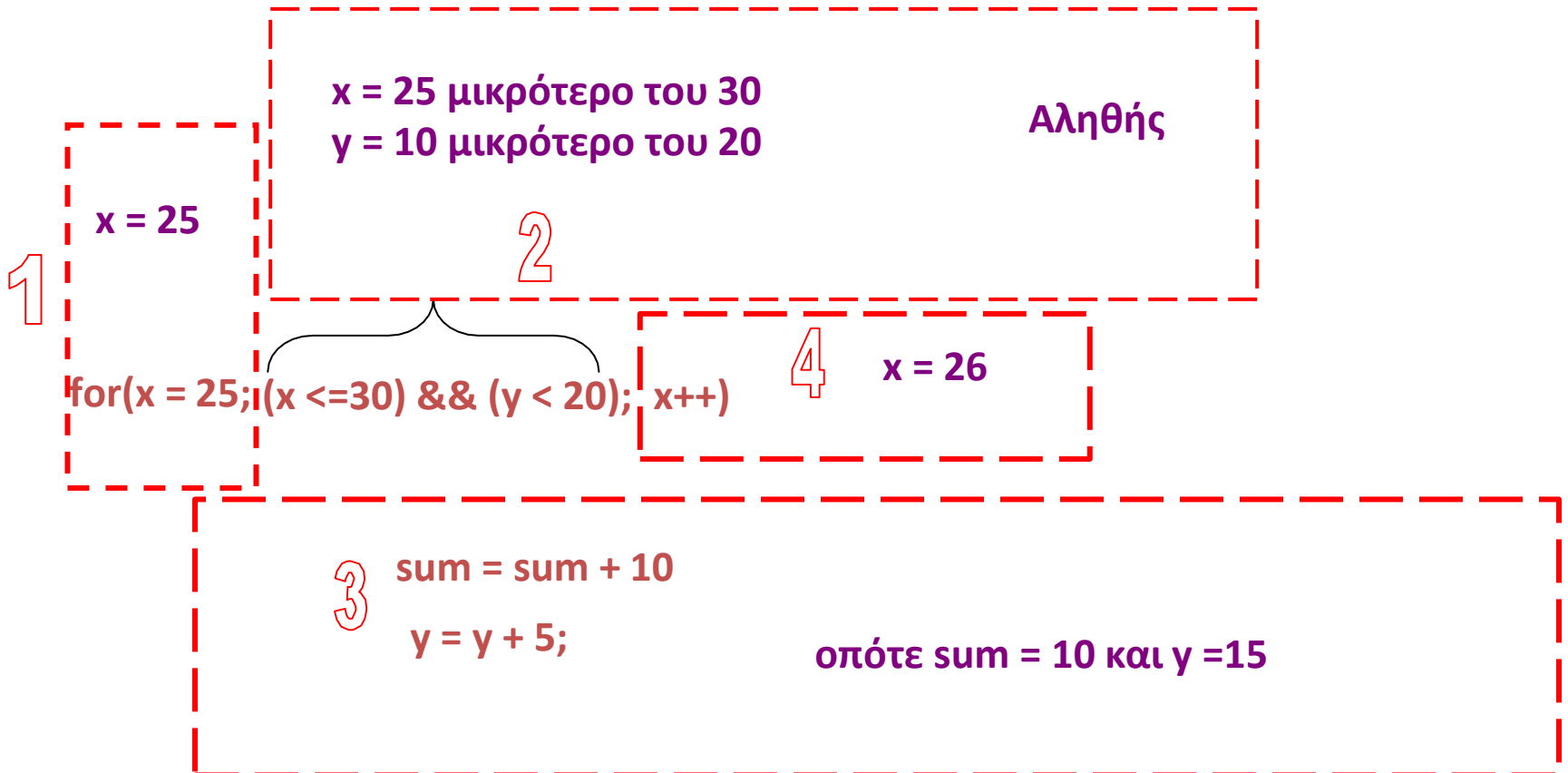
sum = 0

x = 0

Πριν την επανάληψη

y = 10

Πρώτη Επανάληψη



Δεύτερη Επανάληψη

$x = 26$ μικρότερο του 30
 $y = 15$ μικρότερο του 20

Αληθής

2

for($x = 25$; ($x \leq 30$) && ($y < 20$); $x++$)

4

$x = 27$

3

$sum = sum + 10$

$y = y + 5$;

οπότε $sum = 20$ και $y = 20$

Τρίτη Επανάληψη

Ψευδής

$x = 26$ μικρότερο του 30

$y = 20$ μικρότερο του 20

2

for($x = 25$; ($x \leq 30$) && ($y < 20$); $x++$)

Ατελείωτες επαναλήψεις

Μη προκαθορισμένη επανάληψη for

```
for(;;) {  
    εντολές;  
}
```

Οι εντολές break και continue

Η break τερματίζει την επανάληψη

Η continue παρακάμπτει όλες τις εντολές κάτω από αυτήν και συνεχίζει στην αύξηση του βήματος

Το πρόγραμμα εκτυπώνει 7 επειδή το άθροισμα περιλαμβάνει μόνο τους αριθμούς 3 και 4.

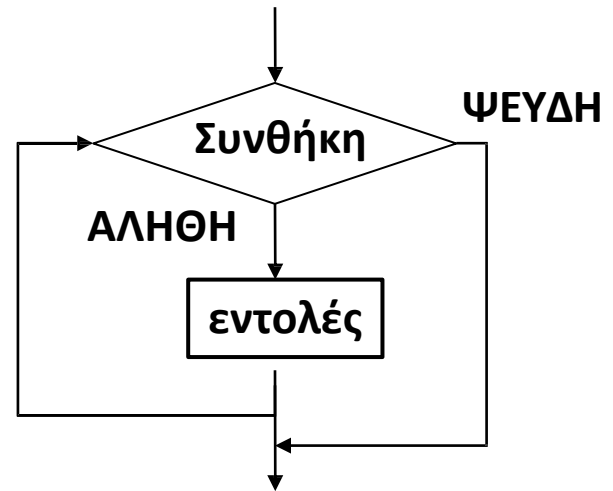
```
public class BreakContinue {  
    public static void main(String[] args) {  
        int i, sum = 0;  
        for(i = 1; i <= 6; i++)  
        {  
            if( i == 5)  
            {  
                break;  
            }  
            if(i < 3)  
            {  
                continue;  
            }  
  
            sum = sum + i;  
        }  
        System.out.println("Sum = " + sum);  
    }  
}
```

Η δήλωση while

Η δήλωση **while** όπως και η δήλωση **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 **ΨΕΥΔΗΣ** οδηγεί στον τερματισμό της δήλωσης, διαφορετικά αν έχει τιμή **ΑΛΗΘΗΣ** τότε εκτελούνται οι **εντολές**. Με το πέρας της εκτέλεσης επαναλαμβάνεται ο έλεγχος της **Συνθήκης**.

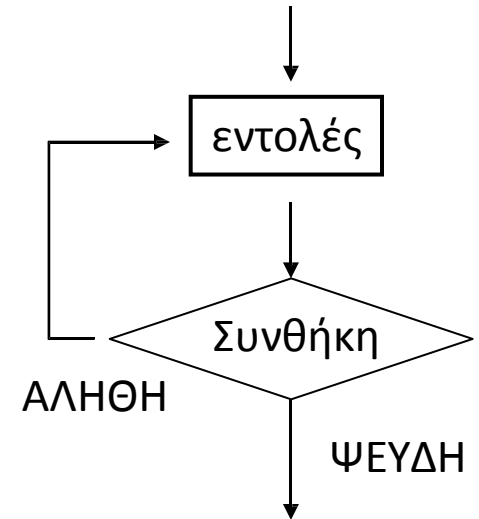


Η δήλωση do.. while

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 **ΨΕΥΔΗΣ** τότε τερματίζεται και η εκτέλεση της δήλωσης. Διαφορετικά αν η συνθήκη έχει τιμή **ΑΛΗΘΗΣ** τότε επαναλαμβάνεται και η εκτέλεση των **εντολών**.



Παράδειγμα do..while

```
public class DoWhileExample {  
    public static void main(String[] args) {  
        int i = 10, sum = 0;  
        do  
        {  
            i = i + 4;  
        }while (i < 50);  
        System.out.println("i = " + i);  
    }  
}
```