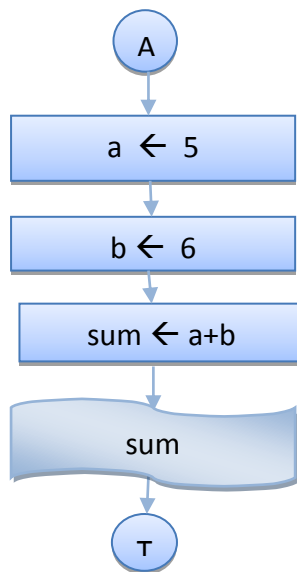


5 ΜΕΘΟΔΟΙ - ΠΑΡΑΜΕΤΡΟΙ

Να γραφεί πρόγραμμα, το οποίο θα δίνει τις τιμές 5 και 6 σε δύο μεταβλητές **a** και **b** και θα υπολογίζει και θα εμφανίζει το άθροισμά τους **sum**.

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ



ΑΛΓΟΡΙΘΜΟΣ

1. Δίνω την τιμή 5 στο **a** (**a ← 5**)
2. Δίνω την τιμή 6 στο **b** (**b ← 6**)
3. Βρίσκω το άθροισμα (**sum ← a+b**)
4. Εμφανίζω την τιμή του **sum**

ΠΡΟΓΡΑΜΜΑ

```
public class SumAB {  
    /* Πρόγραμμα που δίνει τις τιμές 5 και 6 σε 2 ακέραιες μεταβλητές και  
    βρίσκει και εμφανίζει το άθροισμά τους*/  
    public static void main(String[] args) {  
  
        // Δήλωση-Ανάθεση των τιμών 5, 6 στις ακέραιες μεταβλητές a, b  
        int a = 5, b = 6;  
  
        // Δήλωση - Υπολογισμός του αθροίσματος sum  
        int sum = a + b;  
  
        // Εμφάνιση του αθροίσματος sum  
        System.out.println("Άθροισμα = " + sum);  
    }  
}
```

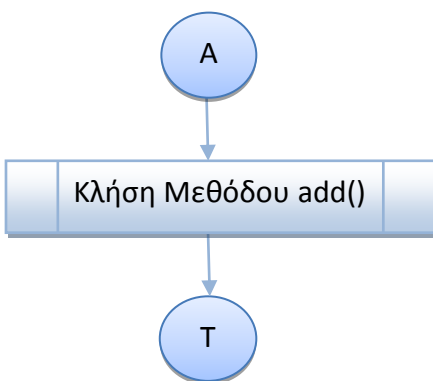
Έξοδος Προγράμματος

Άθροισμα = 11

5.1 ΜΕΘΟΔΟΙ

Θα μπορούσαμε να καλέσουμε μια μέθοδο `add()`, η οποία να δίνει τις τιμές στις μεταβλητές `a`, `b` και να υπολογίζει και να εμφανίζει το άθροισμα `sum` και η μέθοδος `main()` απλώς να την καλεί. Σ' αυτή την περίπτωση το λογικό διάγραμμα και ο αλγόριθμος της `main()` και της μεθόδου `add()` θα γινόταν :

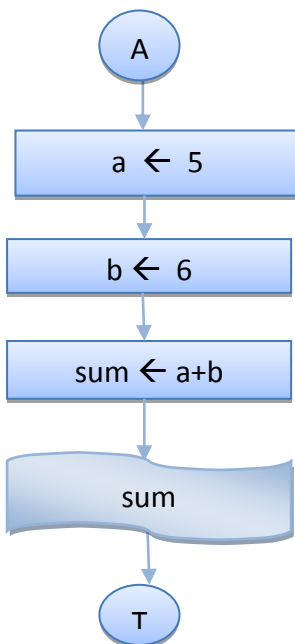
ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ `main()`



ΑΛΓΟΡΙΘΜΟΣ `main()`

1. Κλήση Μεθόδου `add()`

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΜΕΘΟΔΟΥ `add()`



ΑΛΓΟΡΙΘΜΟΣ ΜΕΘΟΔΟΥ `add()`

1. Δίνω την τιμή 5 στο `a` (`a ← 5`)
2. Δίνω την τιμή 6 στο `b` (`b ← 6`)
3. Βρίσκω το άθροισμα `sum` (`sum ← a+b`)
4. Εμφανίζω την τιμή του `sum`

Οι **Μέθοδοι** είναι αυτοτελή τμήματα κώδικα που εκτελούν κάποιες εργασίες και περιέχουν ότι και η μέθοδος `main()` **δηλώσεις** μεταβλητών και **εντολές**. Μια μορφή μεθόδου έχει την παρακάτω σύνταξη :

```
static void <όνομα_μεθόδου>() {  
    εντολές;  
}
```

ΠΡΟΓΡΑΜΜΑ

```
public class MethodsAdd {  
    /* Πρόγραμμα που καλεί τη μέθοδο add(), η οποία δίνει τις τιμές 5 και 6  
    σε 2 ακέραιες μεταβλητές και βρίσκει και εμφανίζει το άθροισμά τους  
    */  
    static void add() {  
  
        // Δήλωση-Ανάθεση των τιμών 5, 6 στις ακέραιες μεταβλητές a, b  
        int a = 5, b = 6;  
  
        // Δήλωση - Υπολογισμός του αθροίσματος sum  
        int sum = a + b;  
  
        // Εμφάνιση του αθροίσματος sum  
        System.out.println("Άθροισμα με κλήση της add() = " + sum);  
    }  
  
    public static void main(String[] args) {  
  
        // Κλήση Μεθόδου add()  
        add();  
    }  
}
```

Έξοδος Προγράμματος

Άθροισμα με κλήση της `add()` = 11

Παρατηρήσεις

- Η μέθοδος `add()` δεν παίρνει **καμία** πληροφορία από την `main()` .
- Δεν στέλνει **καμιά** πληροφορία στην `main()`, γι' αυτό και δηλώνεται τύπου `void`.
- Το `static` χρειάζεται, γιατί δεν είναι μέθοδος κλάσης αντικειμένων.
- Ο κώδικας της μεθόδου `add()` βρίσκεται στην ίδια κλάση με τη μέθοδο `main()` και η κλήση της γίνεται χρησιμοποιώντας απλώς το όνομά της.
- Οι μεταβλητών της μεθόδου λέγονται **τοπικές μεταβλητές**, δημιουργούνται όταν καλείται η μέθοδος, **δεν είναι ορατές στην καλούσα μέθοδο** και παύουν να υπάρχουν, όταν ολοκληρωθεί η εκτέλεση των εντολών της μεθόδου.

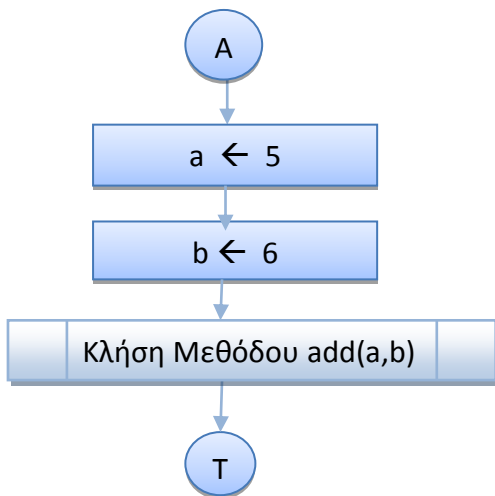
5.1.1 Μέθοδος που Δέχεται Παραμέτρους από τη main()

Η προηγούμενη μέθοδος κάνει μια πολύ συγκεκριμένη δουλειά. Υπολογίζει το άθροισμα των μεταβλητών **a** και **b** για τις συγκεκριμένες τιμές 5 και 6. Αν θέλουμε να υπολογίζει και να εμφανίζει το άθροισμα οποιονδήποτε 2 αριθμών **a** και **b** που θα παίρνουν τιμές στην `main()` θα πρέπει να περάσουμε τις τιμές των μεταβλητών **a** και **b** σαν **παραμέτρους** στη μέθοδο. Αυτό γίνεται στη δήλωση της μεθόδου, οπότε η γενική της σύνταξη θα είναι :

```
static void <όνομα_μεθόδου>(<λίστα_παραμέτρων>) {  
    εντολές;  
}
```

όπου η <λίστα_παραμέτρων> περιλαμβάνει δηλώσεις μεταβλητών χωρισμένες με κόμμα, μέσω των οποίων **περνάμε** στη μέθοδο τις **τιμές** της μεθόδου που την καλεί.

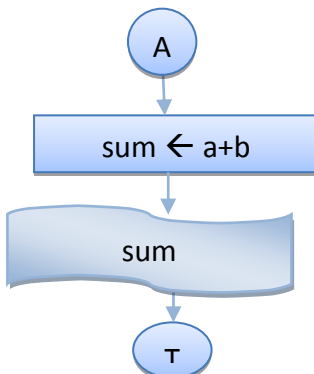
ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ main()



ΑΛΓΟΡΙΘΜΟΣ main()

1. Δίνω την τιμή 5 στο a (**a ← 5**)
2. Δίνω την τιμή 6 στο b (**b ← 6**)
3. Κλήση Μεθόδου `add(a, b)`

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΜΕΘΟΔΟΥ add(a, b)



ΑΛΓΟΡΙΘΜΟΣ ΜΕΘΟΔΟΥ add(a, b)

1. Βρίσκω το άθροισμα (**sum ← a+b**)
2. Εμφανίζω την τιμή του **sum**

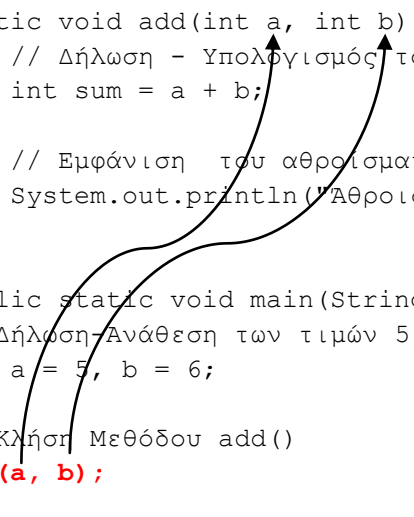
ΠΡΟΓΡΑΜΜΑ

```
public class MethodsAddAB {
/* Πρόγραμμα που καλεί τη μέθοδο add(), η οποία δίνει τις τιμές 5 και 6
σε 2 ακέραιες μεταβλητές και βρίσκει και εμφανίζει το άθροισμά τους
*/
    static void add(int a, int b) {
        // Δήλωση - Υπολογισμός του αθροίσματος sum
        int sum = a + b;

        // Εμφάνιση του αθροίσματος sum
        System.out.println("Άθροισμα με κλήση της add(a, b) = " + sum);
    }

    public static void main(String[] args) {
        // Δήλωση-Ανάθεση των τιμών 5, 6 στις ακέραιες μεταβλητές a, b
        int a = 5, b = 6;

        // Κλήση Μεθόδου add()
        add(a, b);
    }
}
```

A diagram with two arrows. One arrow starts from the `add(a, b);` line in the `main` method and points to the `add` method signature. The second arrow starts from the `add` method body and points back to the `add(a, b);` line in the `main` method, indicating a return path.

Έξοδος Προγράμματος

Άθροισμα με κλήση της `add(a, b) = 11`

Παρατηρήσεις

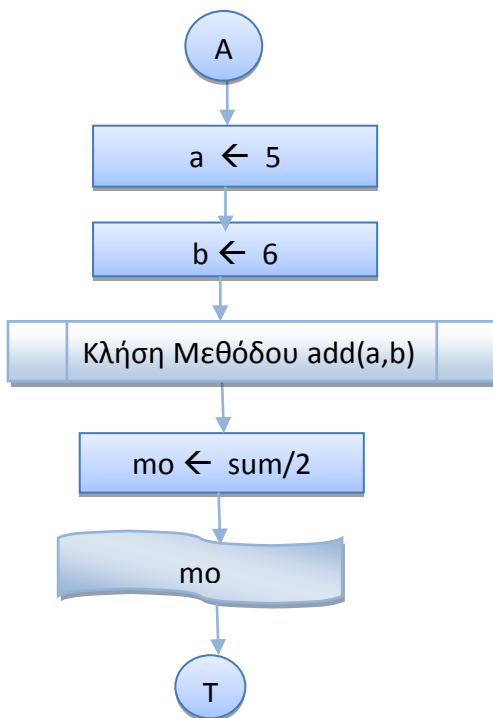
- Οι πληροφορίες περνάνε στη μέθοδο μέσω των μεταβλητών `a` και `b` στη δήλωση.
- Οι μεταβλητές `a` και `b` στη δήλωση της μεθόδου λέγονται **παράμετροι** και μπορούν να έχουν **διαφορετικά** ονόματα από τις μεταβλητές της `main()`.
- Μπορεί να χρησιμοποιηθεί **οποιοδήποτε όνομα** για τις παραμέτρους, αρκεί να χρησιμοποιούνται τα ίδια ονόματα και μέσα στη μέθοδο.
- Πρέπει να συμφωνεί ο **αριθμός**, η **σειρά** και ο **τύπος** των **ορισμάτων** (μεταβλητών της καλούσας μεθόδου) και των **παραμέτρων** της μεθόδου.
- Στην κλήση της μεθόδου **δε χρειάζονται οι τύποι** των μεταβλητών- ορισμάτων.
- Στην κλήση της μεθόδου τα ονόματα των παραμέτρων αντικαθίστανται από τα ονόματα των μεταβλητών της καλούσας μεθόδου. Οι τιμές των μεταβλητών (ορίσματα) περνάνε στην μέθοδο.
- Η μέθοδος **τελειώνει** όταν εκτελεστούν όλες οι εντολές ή αν υπάρχει η εντολή `return`.
- Μπορεί να μη χρησιμοποιηθεί η τοπική μεταβλητή `sum`. Σ' αυτή την περίπτωση, ο κώδικας της μεθόδου θα γίνει :

```
static void add(int a, int b) {
    // Υπολογισμός - Εμφάνιση του αθροίσματος (a + b)
    System.out.println("Άθροισμα = " + (a + b));
}
```

5.1.2 Μέθοδος που Επιστρέφει και το Άθροισμα στη main ()

Να γραφεί πρόγραμμα, το οποίο θα δίνει τις τιμές 5 και 6 σε δύο μεταβλητές **a** και **b** και θα καλεί τη μέθοδο `add()`, η οποία υπολογίζει και **επιστρέφει** στη `main()` την τιμή του αθροίσματος **sum**, την οποία και θα εμφανίζει. Μετά το πρόγραμμα θα υπολογίζει και θα εμφανίζει και το μέσο όρο **mo**.

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ main()



ΑΛΓΟΡΙΘΜΟΣ main()

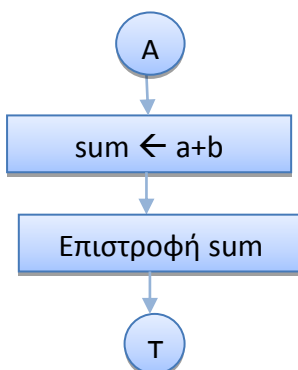
1. Δίνω την τιμή 5 στο **a** (**a** ← 5)
2. Δίνω την τιμή 6 στο **b** (**b** ← 6)
3. Κλήση Μεθόδου `add(a, b)`
4. Υπολογισμός Μέσου Όρου (**mo** ← **sum/2**)
5. Εμφάνιση Μέσου Όρου **mo**

ΑΛΓΟΡΙΘΜΟΣ ΜΕΘΟΔΟΥ

`add(a, b)`

1. Βρίσκω το άθροισμα (**sum** ← **a+b**)
2. **Επιστρέφω** το άθροισμα **sum**

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΜΕΘΟΔΟΥ add(a, b)



Για να μπορέσει μια μέθοδος να **επιστρέψει** μια τιμή στην καλούσα μέθοδο, θα πρέπει :

- Να δηλωθεί στην **επικεφαλίδα** της μεθόδου ο **τύπος** της τιμής που θα επιστραφεί.
- Να υπάρχει **μέσα** στον κώδικα της μεθόδου η εντολή **return** με την οποία θα επιστραφεί αυτή η τιμή.

Η σύνταξη της δήλωσης της μεθόδου θα έχει τη μορφή :

```
static <τύπος_επιστροφής> <όνομα_μεθόδου> (<λίστα_παραμέτρων>) {  
    εντολές;  
    return <έκφραση_τύπου_επιστροφής>;  
}
```

ΠΡΟΓΡΑΜΜΑ

```
public class MethodsIntAddAB {  
    /*
```

Πρόγραμμα το οποίο δίνει τις τιμές 5 και 6 σε δύο μεταβλητές **a** και **b** και καλεί τη μέθοδο `add()`, η οποία υπολογίζει και **επιστρέφει** στη `main()` την τιμή του αθροίσματος **sum**, την οποία και εμφανίζει. Μετά το πρόγραμμα υπολογίζει και εμφανίζει και το μέσο όρο **mo**.

```
*/
```

```
    static int add(int a, int b) {  
  
        // Δήλωση - Υπολογισμός του αθροίσματος sum  
        int sum = a + b;  
  
        // Επιστροφή Τιμής Αθροίσματος sum  
        return sum;  
    }  
  
    public static void main(String[] args) {  
  
        // Δήλωση-Ανάθεση των τιμών 5, 6 στις ακέραιες μεταβλητές a, b  
        int a = 5, b = 6;  
  
        // Κλήση Μεθόδου add()  
        int sum = add(a, b);  
  
        // Δήλωση - Υπολογισμός του Μέσου Όρου mo  
        double mo = (double) sum/2;  
  
        // Εμφάνιση του αθροίσματος sum και του Μέσου Όρου mo  
        System.out.println("Αθροισμα = " + sum + " Μέσος Όρος = " + mo );  
    }  
}
```

Έξοδος Προγράμματος

```
run:  
Αθροισμα = 11 Μέσος Όρος = 5.5  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Παρατηρήσεις

- Όταν επιστρέφει η τιμή μιας μεταβλητής από κάποια μέθοδο πρέπει να δηλώνεται ο **τύπος** της.
- Ο **τύπος της μεθόδου** πρέπει να είναι ίδιος με τον **τύπο** της **μεταβλητής** που επιστρέφει.
- Η **κλήση** της μεθόδου χρησιμοποιείται στην καλούσα μέθοδο σαν μια οποιαδήποτε μεταβλητή σε εντολές εκχώρησης, ελέγχου, εμφάνισης κ.λ.π..
- Η μέθοδος επιστρέφει **μόνο μια τιμή** για **μεταβλητές**, εκφράσεις **απλού τύπου**.
- Μπορεί να υπάρχουν στη μέθοδο **περισσότερες** από μία εντολές **return**.
- Μπορεί να μη χρησιμοποιηθεί η τοπική μεταβλητή `sum` και να επιστρέψει η έκφραση υπολογισμού του. Σ' αυτή την περίπτωση ο κώδικας της μεθόδου θα είναι :

```
static int add(int a, int b) {  
  
    // Υπολογισμός - Επιστροφή Τιμής Αθροίσματος (a + b)  
    return (a + b);  
}
```

- Μπορεί να μην χρησιμοποιηθεί η μεταβλητή `sum` στη `main()` και να χρησιμοποιηθεί το όνομα της μεθόδου με τις παραμέτρους. Αντί των εντολών :

```
// Κλήση Μεθόδου add()  
int sum = add(a, b);  
  
// Δήλωση - Υπολογισμός του Μέσου Όρου mo  
double mo = (double) sum/2;
```

θα μπορούσαμε να γράψουμε :

```
// Δήλωση - Υπολογισμός του Μέσου Όρου mo - Κλήση Μεθόδου add()  
double mo = (double) (add(a, b))/2;
```

- Ο τύπος επιστροφής μιας μεθόδου μπορεί να είναι οποιοσδήποτε από τους απλούς τύπους, ακέραιος (`byte`, `short`, `int`, `long`), κινητής υποδιαστολής (`float`, `double`), χαρακτήρας (`char`), ή λογικός (`boolean`) που επιστρέφει τις τιμές `true` ή `false`, όπως στο επόμενο παράδειγμα :

5.1.2 Μέθοδος που Επιστρέφει Αποτέλεσμα Τύπου boolean

Να γραφεί πρόγραμμα, το οποίο θα δημιουργεί έναν τυχαίο ακέραιο **a** στο [1, 10], θα καλεί μια μέθοδο, η οποία θα βρίσκει αν ο αριθμός είναι ΑΡΤΙΟΣ ή ΠΕΡΙΤΤΟΣ και θα εμφανίζει το κατάλληλο μήνυμα.

ΠΡΟΓΡΑΜΜΑ

```
public class BooleanArtios {
    /*
    Πρόγραμμα, το οποίο θα δημιουργεί έναν τυχαίο ακέραιο a στο [1, 10], καλεί
    μια μέθοδο, η οποία βρίσκει αν ο αριθμός είναι ΑΡΤΙΟΣ ή ΠΕΡΙΤΤΟΣ και
    εμφανίζει το κατάλληλο μήνυμα
    */
    static boolean isArtios(int a){
        boolean artios = false; // Αρχική τιμή false
        if ( a%2 == 0) // Ο a διαιρείται ακριβώς με το 2
            artios = true; // Είναι ΑΡΤΙΟΣ
        return artios; // Επιστροφή Boolean τιμής
    }
    public static void main(String[] args) {
        int a = (int)(Math.random()*10) + 1; // Τυχαίος Ακέραιος
        if (isArtios(a)) // Είναι άρτιος ???
            System.out.println("Ο αριθμός " + a + " είναι ΑΡΤΙΟΣ");
        else
            System.out.println("Ο αριθμός " + a + " είναι ΠΕΡΙΤΤΟΣ");
    }
}
```

Έξοδος Προγράμματος

```
run:
Ο αριθμός 6 είναι ΑΡΤΙΟΣ
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
run:
Ο αριθμός 9 είναι ΠΕΡΙΤΤΟΣ
BUILD SUCCESSFUL (total time: 0 seconds)
```

Παρατήρηση

Δε χρειάζεται να ελέγξουμε αν η τιμή που επιστρέφει η μέθοδος **isArtios(a)** είναι **true** ή **false**, δηλαδή η εντολή **if (isArtios(a))** να γίνει :

```
if (isArtios(a)) == true)
```