Langages de programmation

www.mariaiulianadascalu.com
mariaiuliana.dascalu@gmail.com

UPB, FILS

Rappel & aujourd'hui

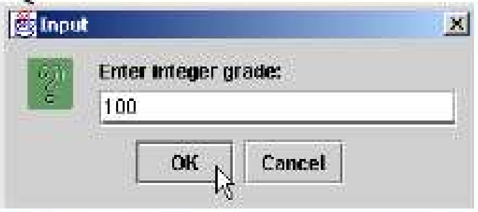
- Flot de contrôle
- Tableaux
- Fonctions

- Aujourd’hui:
  - Tableaux dynamiques et hétérogènes (ArrayList)
  - Passage par val et par ref
  - Fonctions récursives

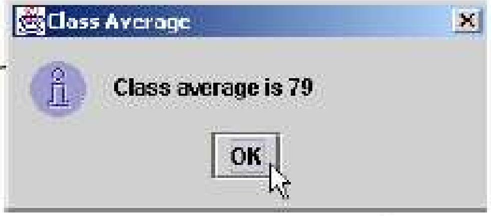
```
// Java extension packages
import javax.swing.JOptionPane;
public class Average1 {
       // main method begins execution of Java application
      public static void main( String args[] ) {
         int total,          // sum of grades input by user
             gradeCounter,   // number of grades entered
             gradeValue,     // grade value
             average;        // average of all grades
            String grade;       // grade typed by user
         // Initialization Phase
         total = 0;          // clear total
         gradeCounter = 1;   // prepare to loop
         // Processing Phase
         while ( gradeCounter <= 10 ) {   // loop 10 times
            // prompt for input and read grade from user
            grade = JOptionPane.showInputDialog("Enter integer grade: " );
            // convert grade from a String to an integer
            gradeValue = Integer.parseInt( grade );
            // add gradeValue to total
            total = total + gradeValue;
            // add 1 to gradeCounter
            gradeCounter = gradeCounter + 1;
         }  // end while structure
         // Termination Phase
         average = total / 10;  // perform integer division
         // display average of exam grades
         JOptionPane.showMessageDialog( null,
            "Class average is " + average, "Class Average",
            JOptionPane.INFORMATION_MESSAGE );
          System.exit( 0 );     // terminate the program
      }  // end method main
   }  // end class Average1
```

Declare variables; gradeCounter is the counter

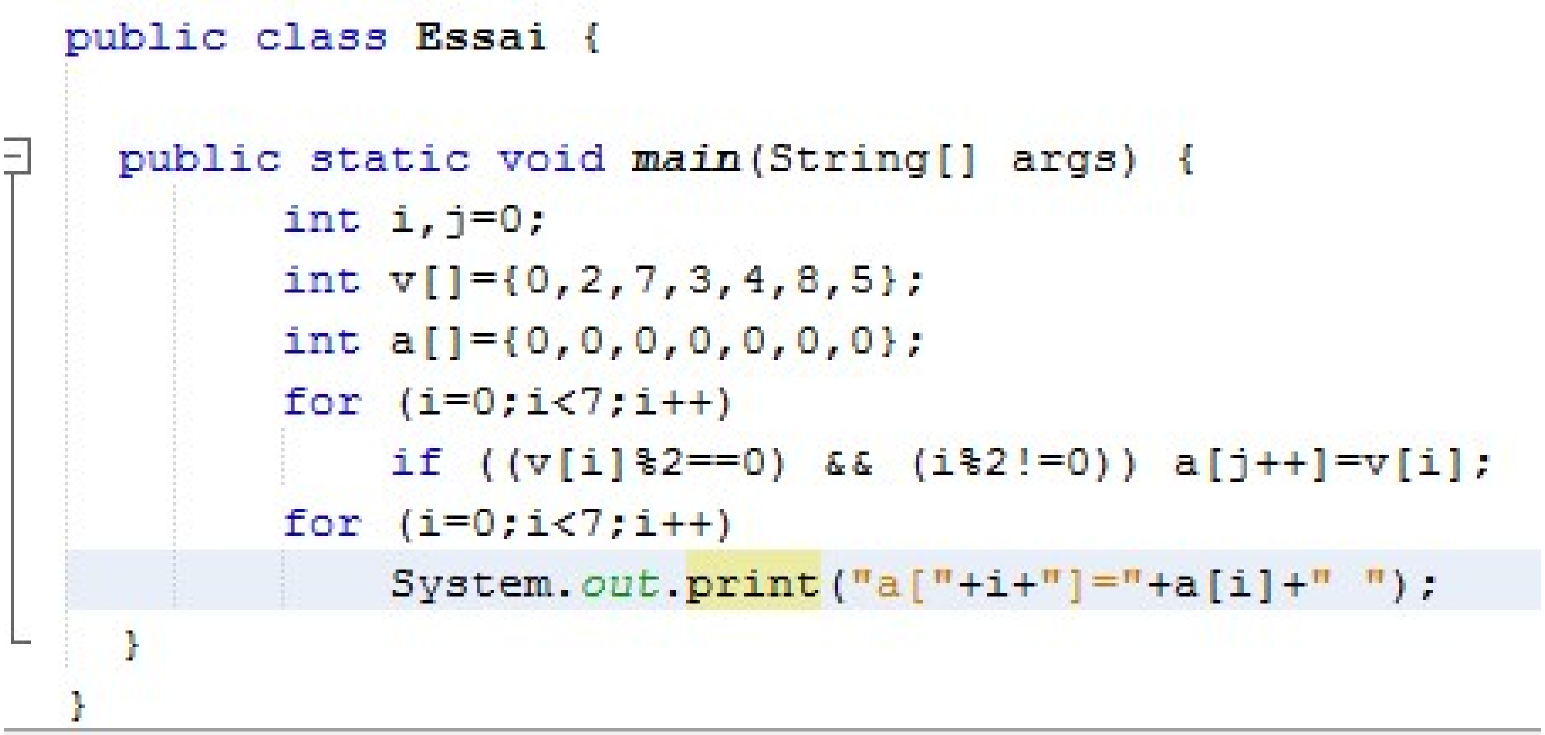
Continue looping as long as gradeCounter is less than or equal to 10



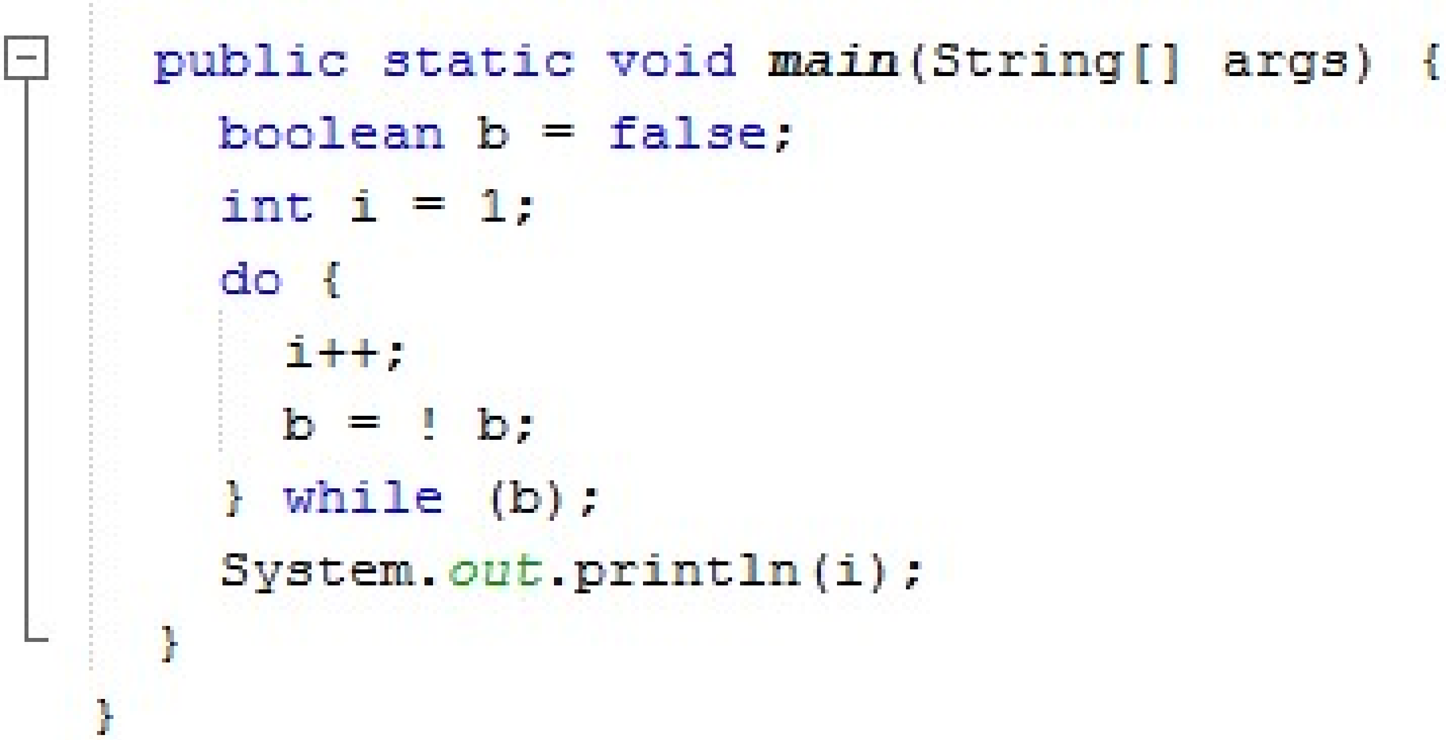
Enter integer grade:
100



Class average is 79



```
public class Essai {

    public static void main(String[] args) {
            int i,j=0;
            int v[]={0,2,7,3,4,8,5};
            int a[]={0,0,0,0,0,0,0};
            for (i=0;i<7;i++)
                if ((v[i]%2==0) && (i%2!=0)) a[j++]=v[i];
            for (i=0;i<7;i++)
                System.out.print("a["+i+"]="+a[i]+" ");
    }
}
```



```
  public static void main(String[] args) {
    boolean b = false;
    int i = 1;
    do {
      i++;
      b = ! b;
    } while (b);
    System.out.println(i);
  }
}
```

```
class Essai{
    static int f(int n){
        int m = n+1;
```

La variable m n'est connue (on dit vue) que par la fonction f. En particulier, on ne peut l'utiliser dans la fonction main ou toute autre fonction qui serait dans la classe.

```
        return 2*m;
    }

    public static void main(String[] args){
        int m = 3;
```

La variable m d'ici est différente de la variable m de la function ci-dessus

```
        System.out.print("résultat=");
        System.out.print(f(4));
        System.out.print(" m=");
        System.out.println(m);
        return;
    }
}
```

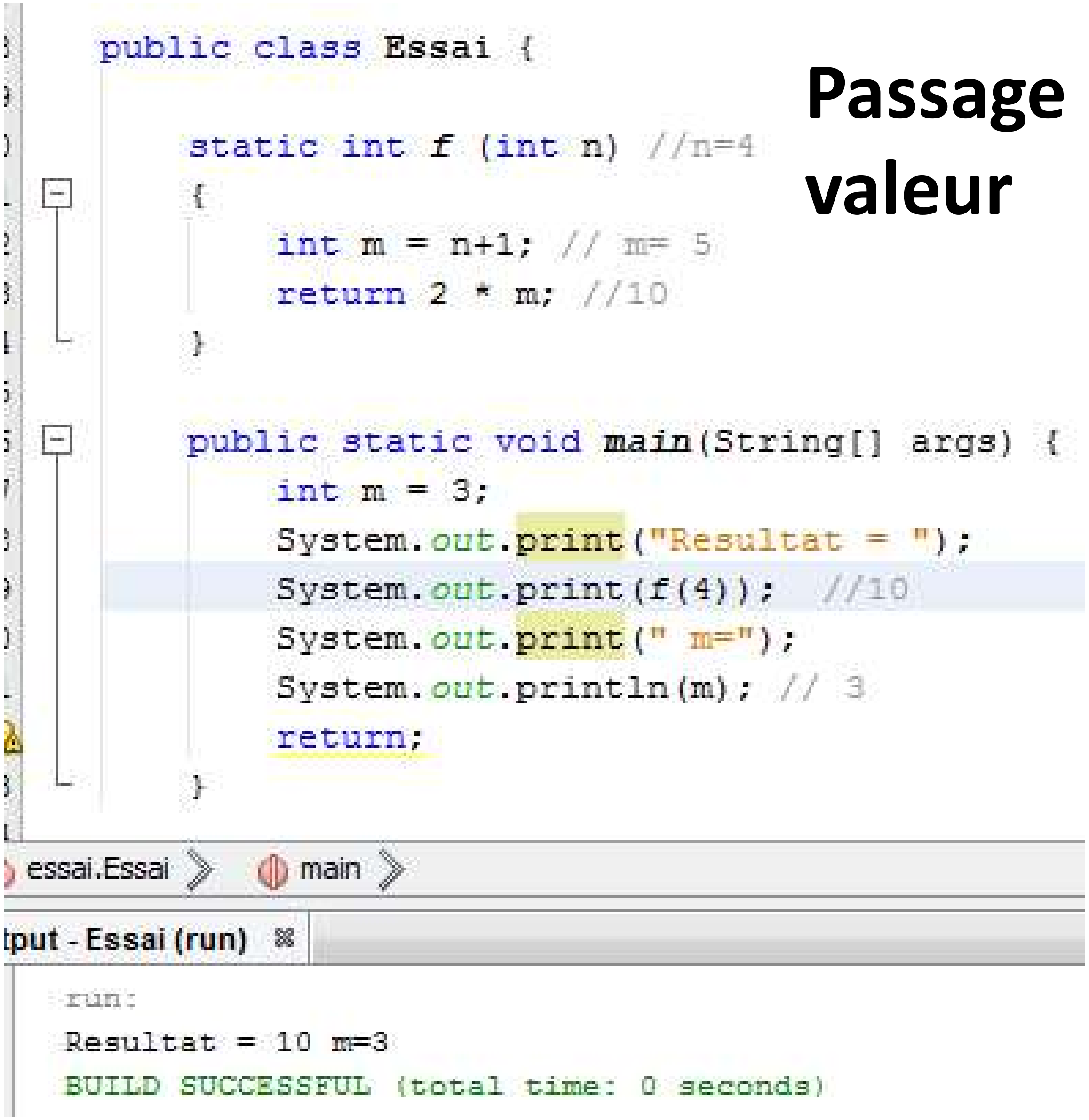
Qu'est-ce qui s'affiche à l'écran ? On a le choix entre :

```
résultat=10 m=5
```

ou

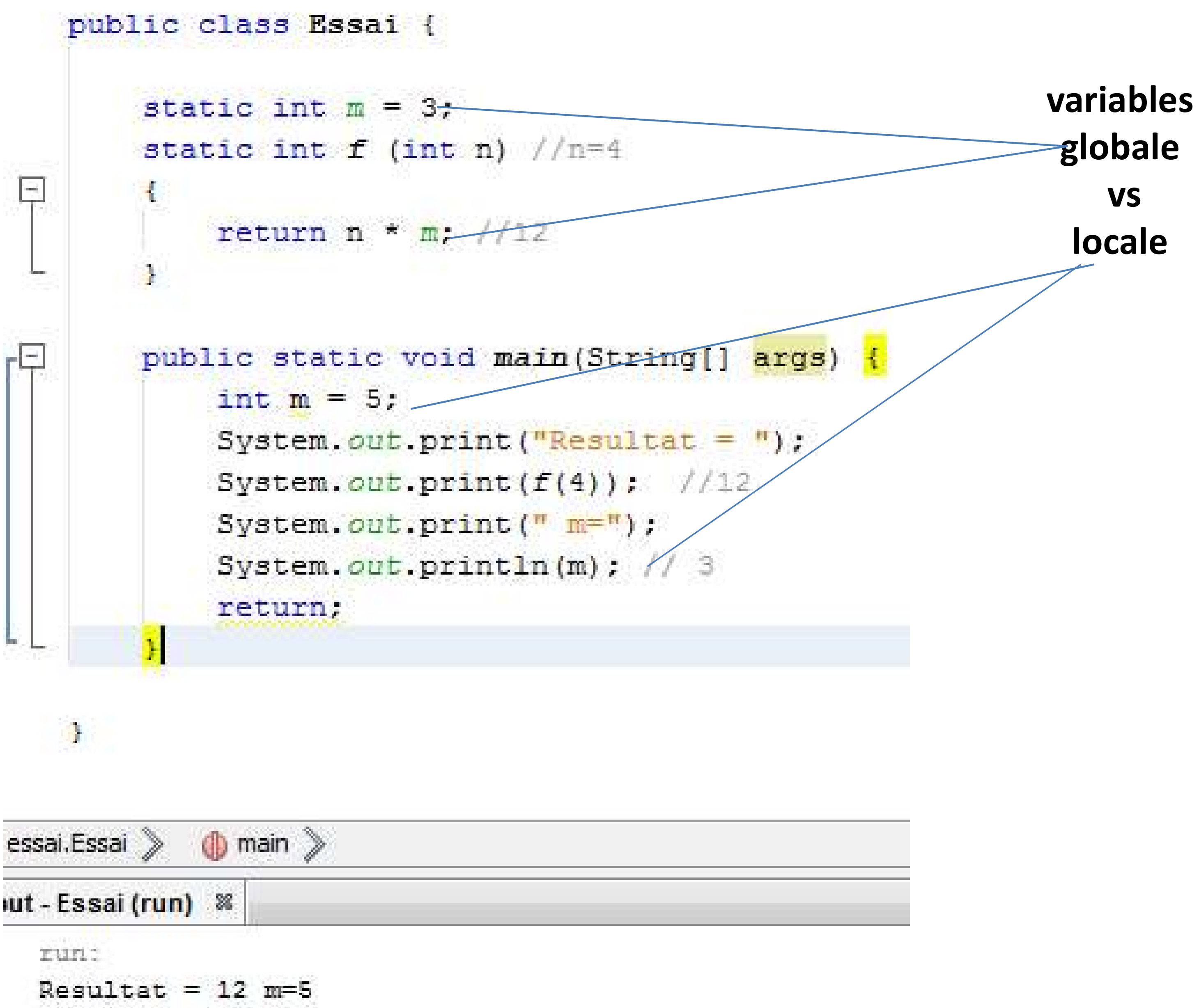
```
résultat=10 m=3
```

Passage par valeur



```
public class Essai {

    static int f (int n) //n=4
    {
        int m = n+1; // m= 5
        return 2 * m; //10
    }

    public static void main(String[] args) {
        int m = 3;
        System.out.print("Resultat = ");
        System.out.print(f(4));  //10
        System.out.print(" m=");
        System.out.println(m); // 3
        return;
    }
```

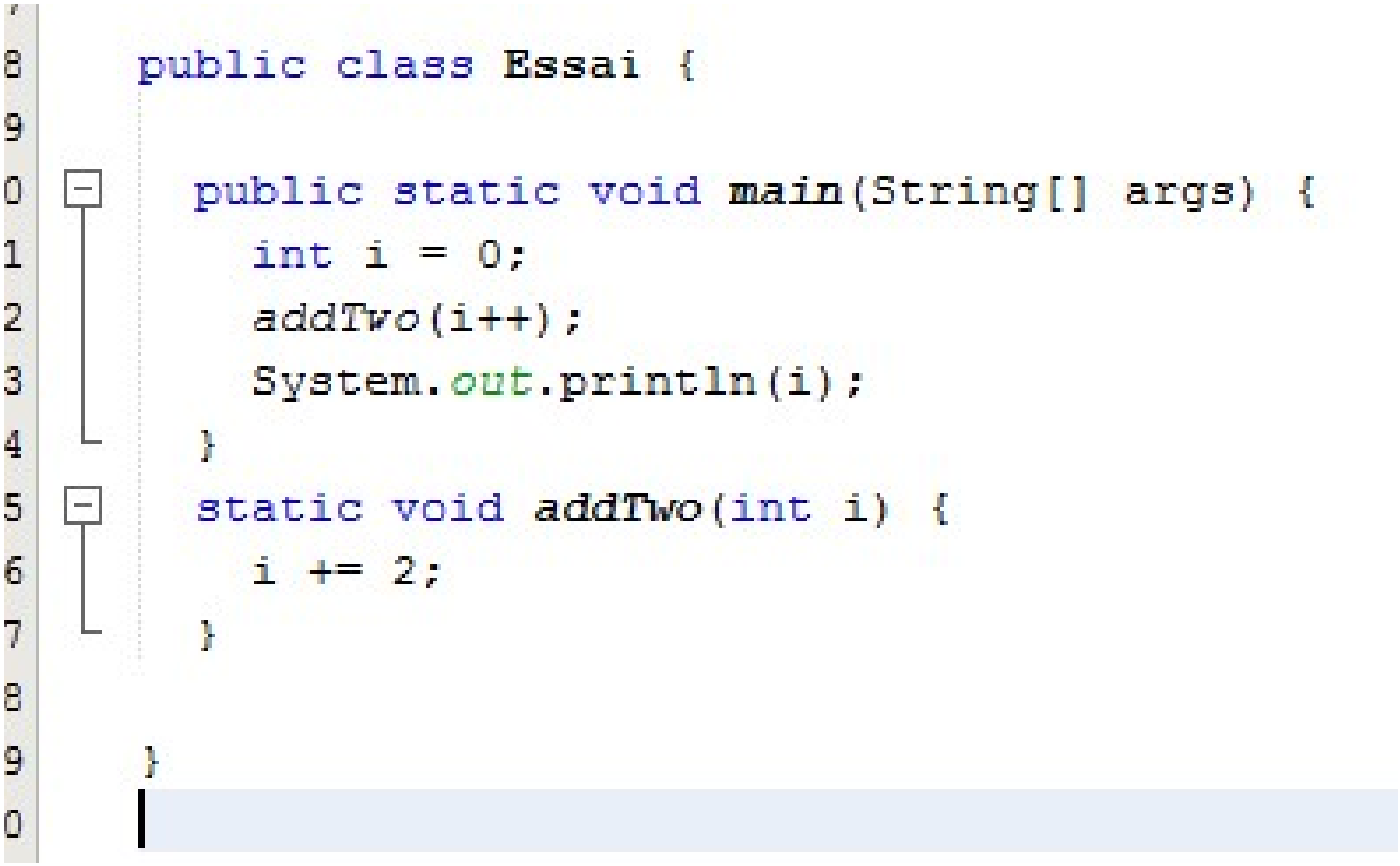
```
run:
Resultat = 10 m=3
BUILD SUCCESSFUL (total time: 0 seconds)
```



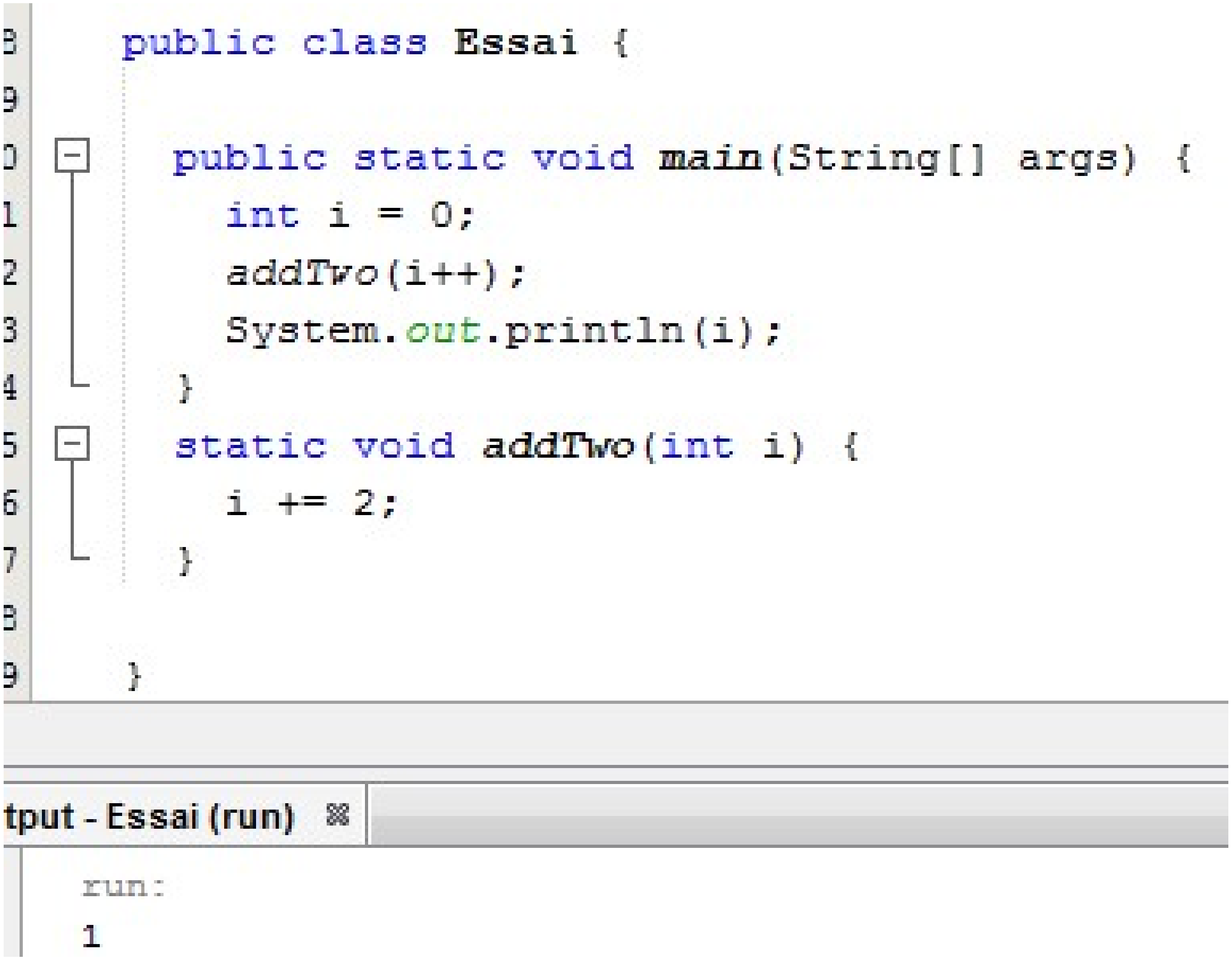
```
public class Essai {

    static int m = 3;
    static int f (int n) //n=4
    {
        return n * m; //12
    }

    public static void main(String[] args) {
        int m = 5;
        System.out.print("Resultat = ");
        System.out.print(f(4));  //12
        System.out.print(" m=");
        System.out.println(m); // 3
        return;
    }

}
```

```
run:
Resultat = 12 m=5
```

variables
globale
vs
locale

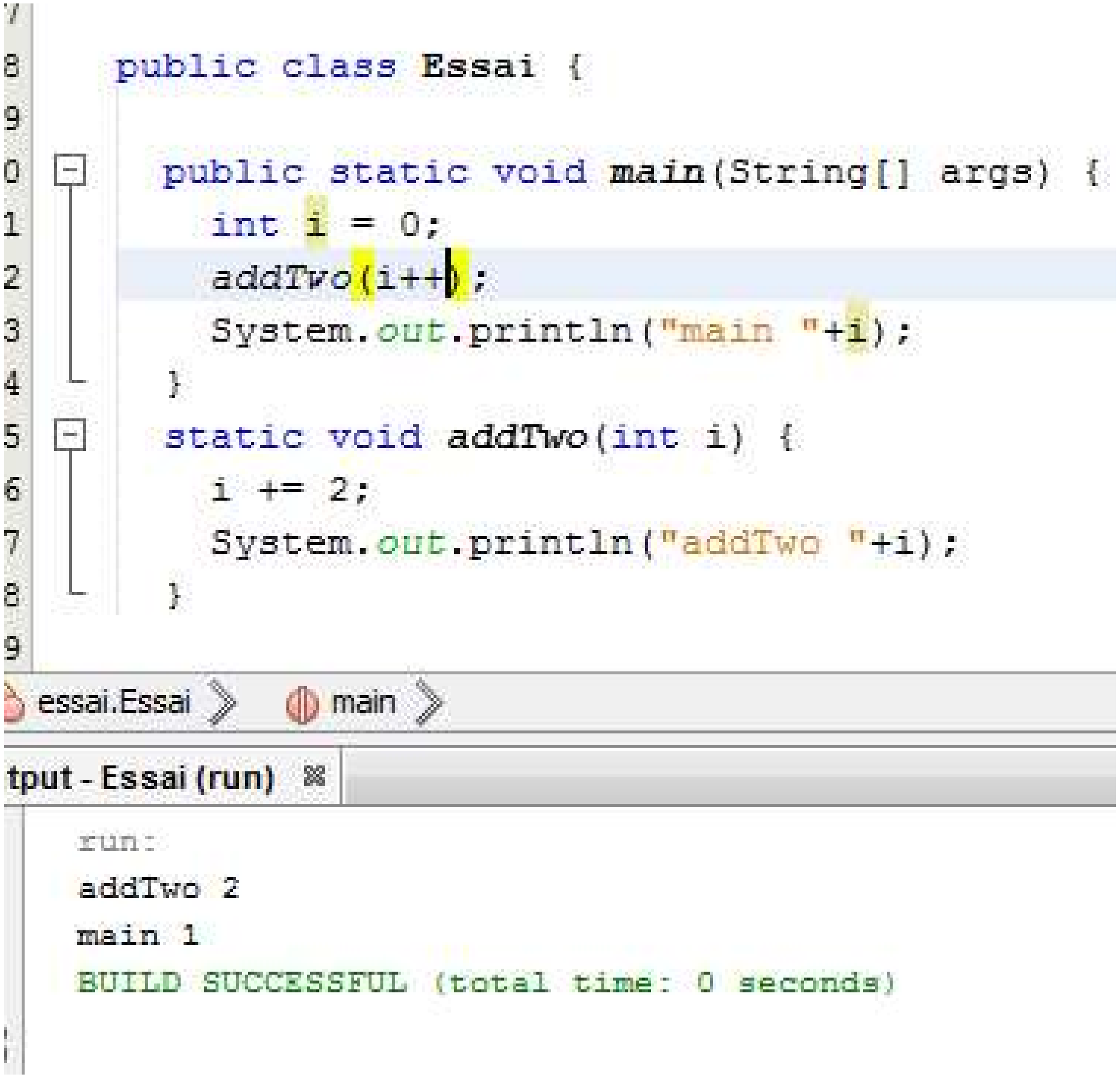


```java
public class Essai {

  public static void main(String[] args) {
    int i = 0;
    addTwo(i++);
    System.out.println(i);
  }
  static void addTwo(int i) {
    i += 2;
  }

}
```



```java
public class Essai {

  public static void main(String[] args) {
    int i = 0;
    addTwo(i++);
    System.out.println(i);
  }
  static void addTwo(int i) {
    i += 2;
  }

}
```

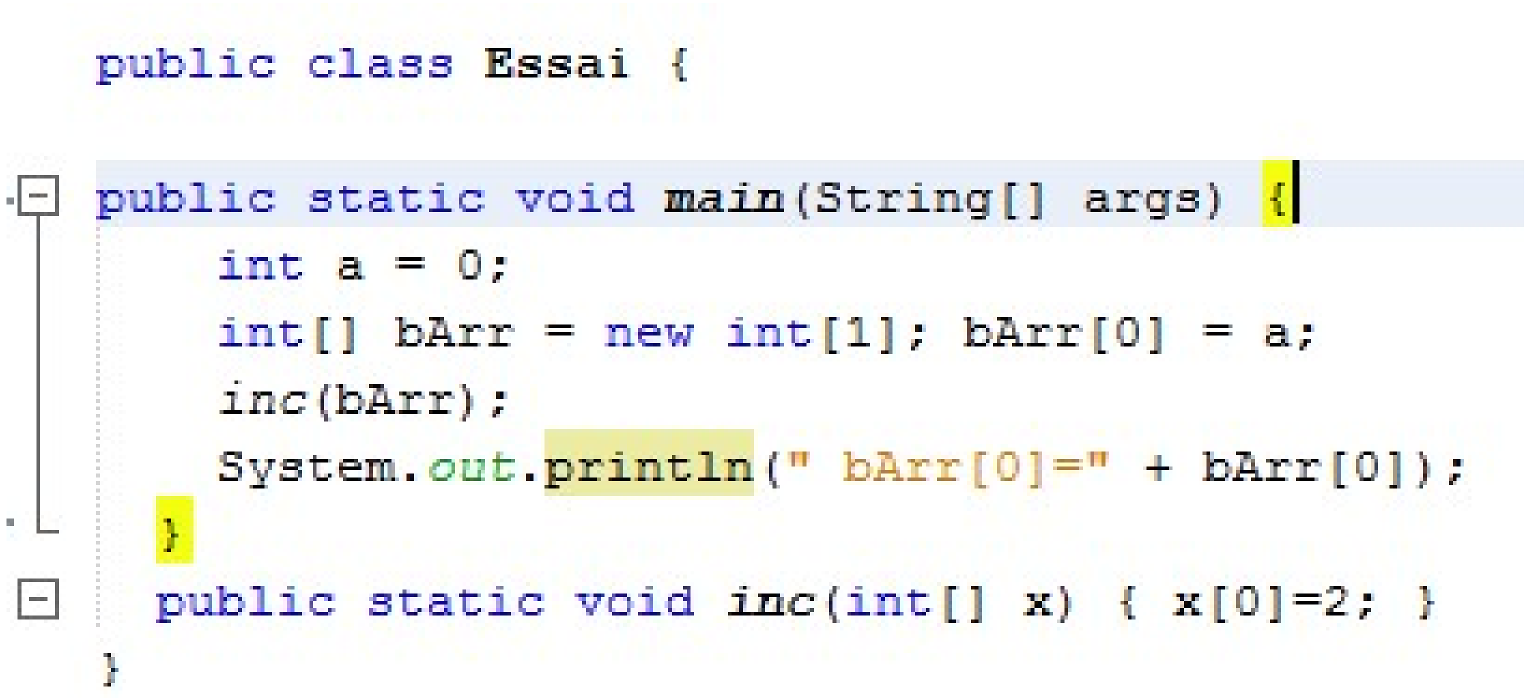
```
run:
1
```



```
public class Essai {

  public static void main(String[] args) {
    int i = 0;
    addTwo(i++);
    System.out.println("main "+i);
  }
  static void addTwo(int i) {
    i += 2;
    System.out.println("addTwo "+i);
  }

```

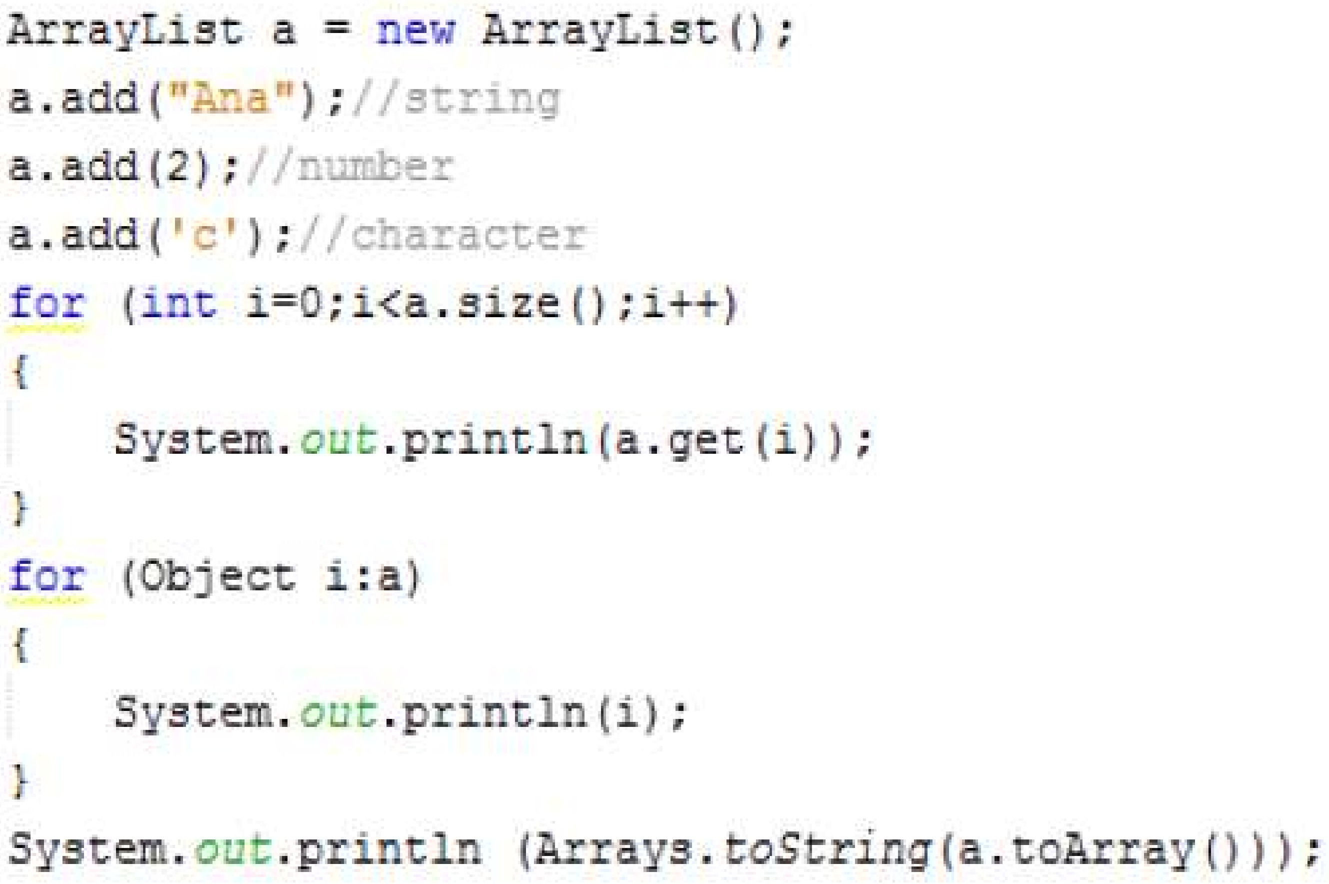
```
run:
addTwo 2
main 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Passage par reference



```
public class Essai {

public static void main(String[] args) {
    int a = 0;
    int[] bArr = new int[1]; bArr[0] = a;
    inc(bArr);
    System.out.println(" bArr[0]=" + bArr[0]);
  }
  public static void inc(int[] x) { x[0]=2; }
}
```

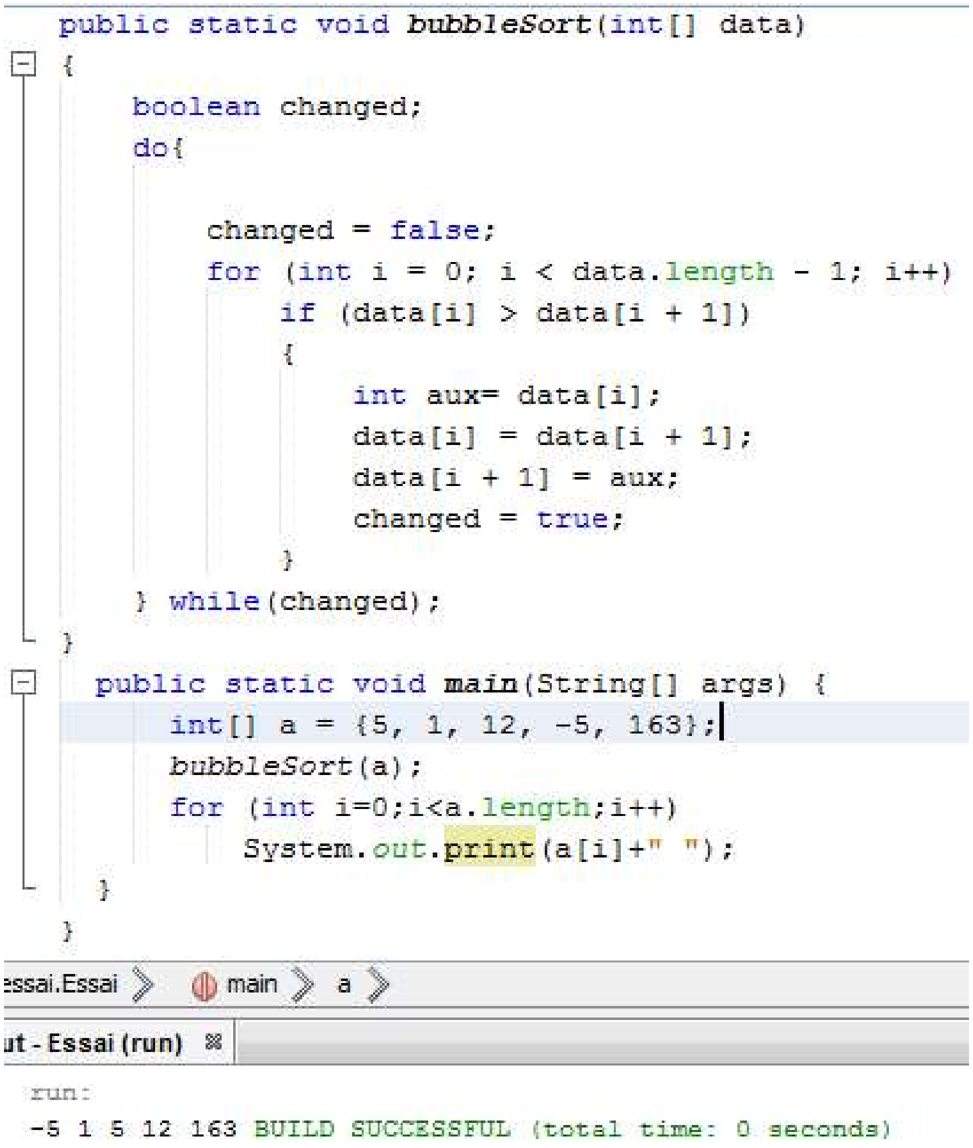
Tableaux dynamiques et hétérogenes (ArrayList)



```
ArrayList a = new ArrayList();
a.add("Ana");//string
a.add(2);//number
a.add('c');//character
for (int i=0;i<a.size();i++)
{
    System.out.println(a.get(i));
}
for (Object i:a)
{
    System.out.println(i);
}
System.out.println (Arrays.toString(a.toArray()));
```

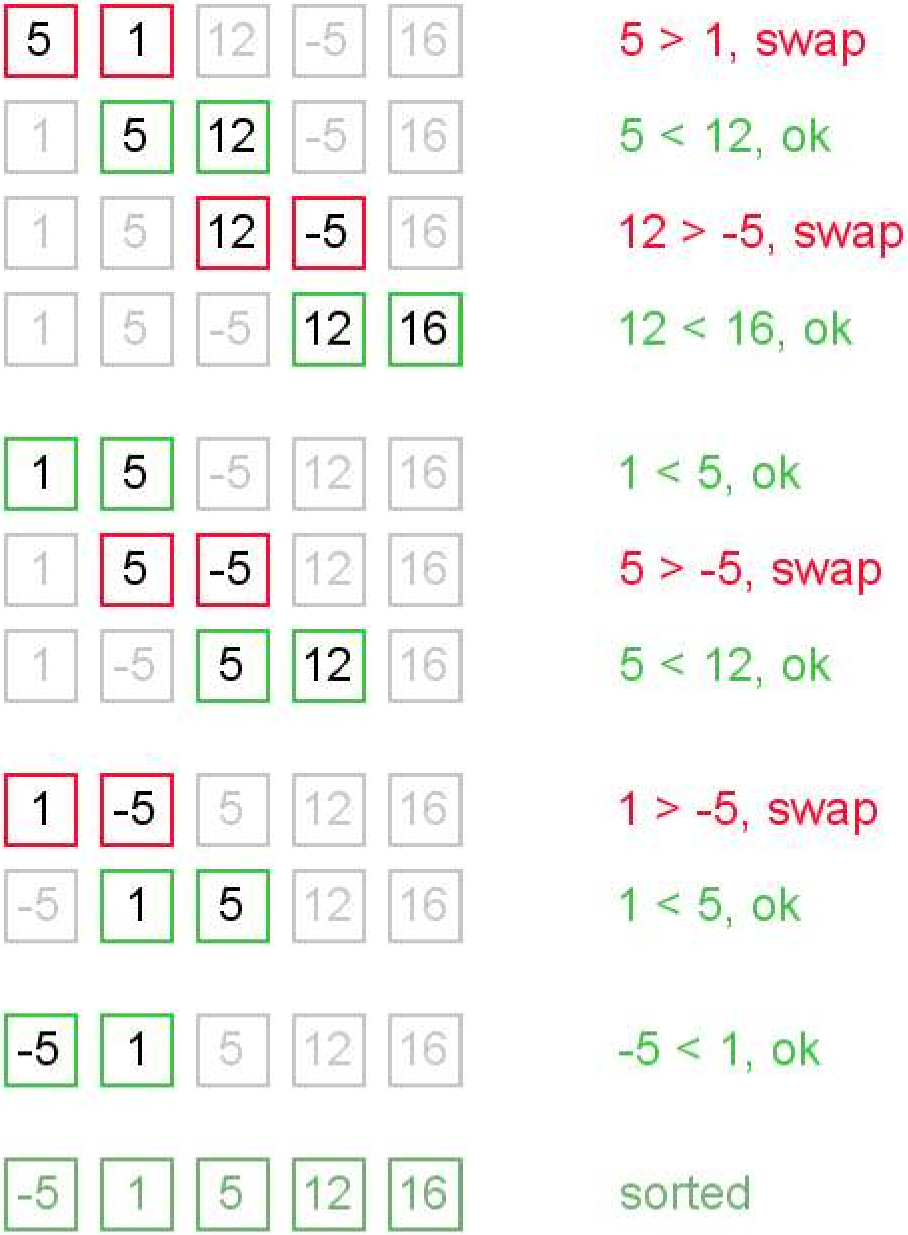
Algorithmes de tri – Bubble Tri

- Algorithmes = ensemble de règles à suivre dans les calculs ou d'autres opérations de résolution de problèmes

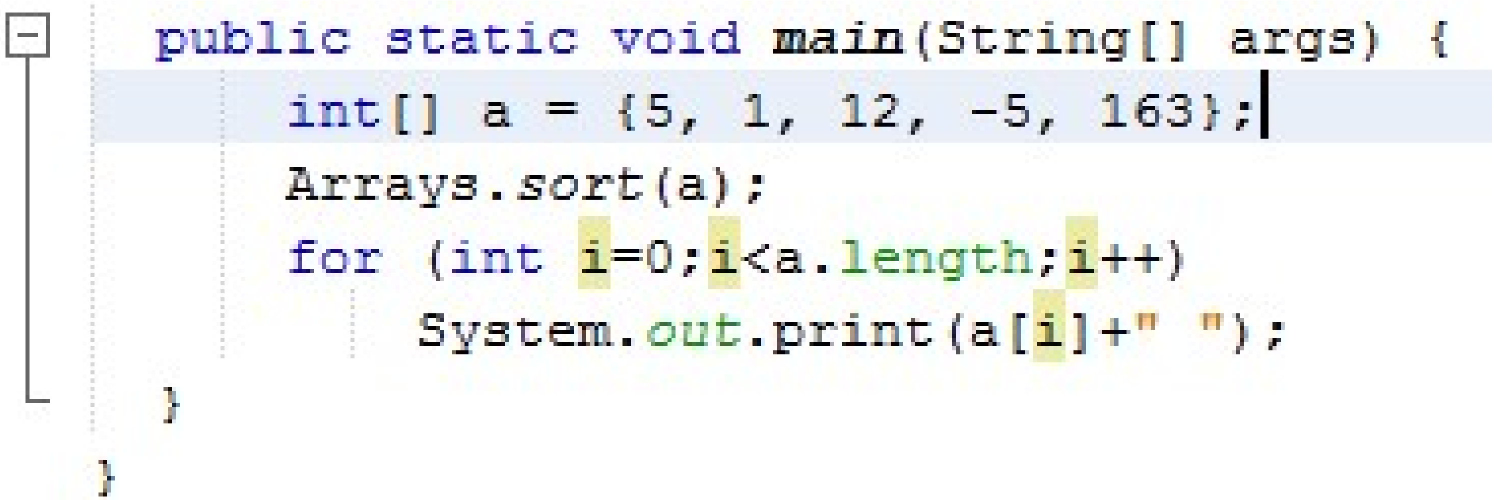


```
public static void bubbleSort(int[] data)
{
    boolean changed;
    do{

        changed = false;
        for (int i = 0; i < data.length - 1; i++)
            if (data[i] > data[i + 1])
            {
                int aux= data[i];
                data[i] = data[i + 1];
                data[i + 1] = aux;
                changed = true;
            }
    } while(changed);
}
  public static void main(String[] args) {
      int[] a = {5, 1, 12, -5, 163};
      bubbleSort(a);
      for (int i=0;i<a.length;i++)
          System.out.print(a[i]+" ");
  }
}
```

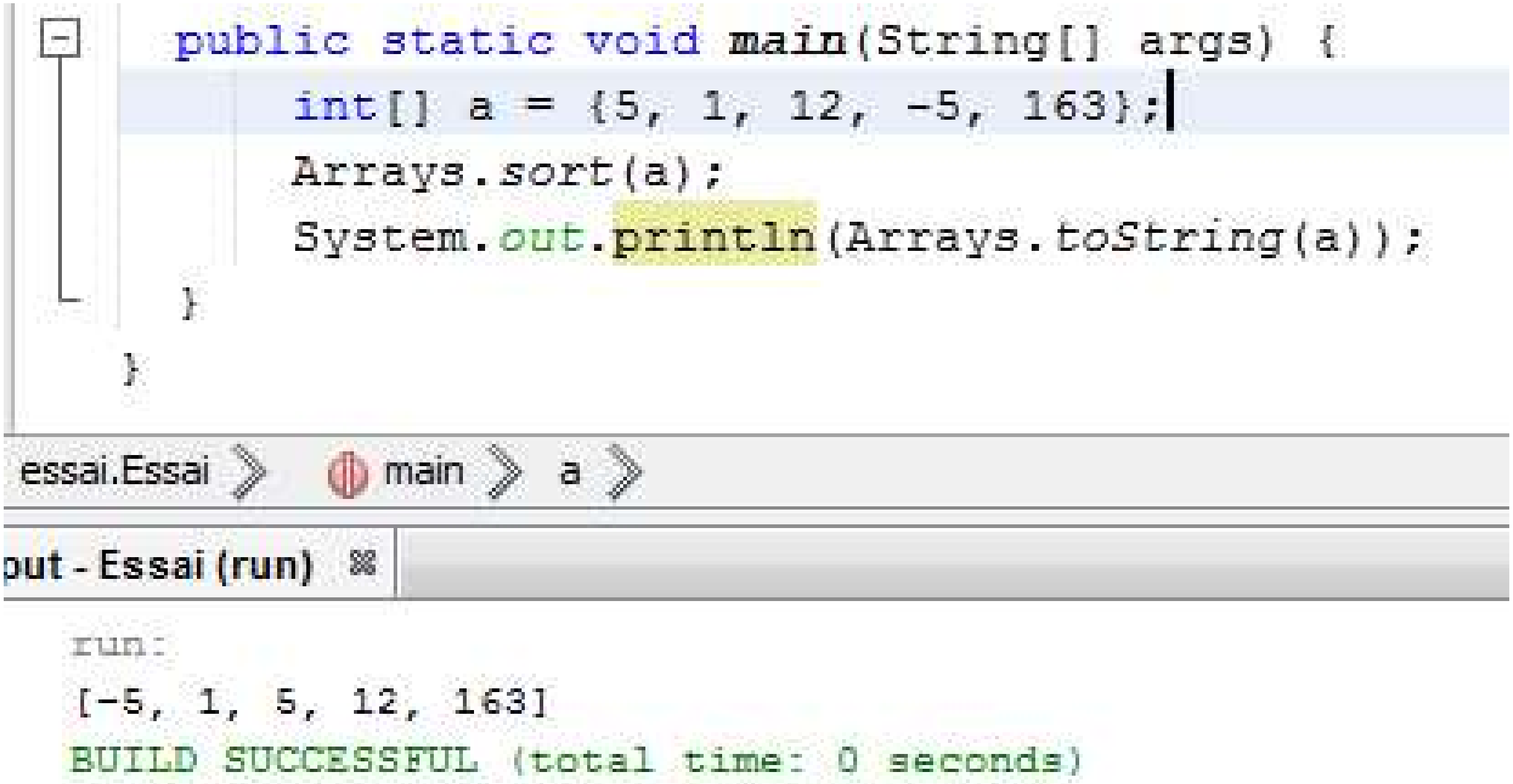
```
run:
-5 1 5 12 163 BUILD SUCCESSFUL (total time: 0 seconds)
```



5	1	12	-5	16	5 > 1, swap
1	5	12	-5	16	5 < 12, ok
1	5	12	-5	16	12 > -5, swap
1	5	-5	12	16	12 < 16, ok
1	5	-5	12	16	1 < 5, ok
1	5	-5	12	16	5 > -5, swap
1	-5	5	12	16	5 < 12, ok
1	-5	5	12	16	1 > -5, swap
-5	1	5	12	16	1 < 5, ok
-5	1	5	12	16	-5 < 1, ok
-5	1	5	12	16	sorted



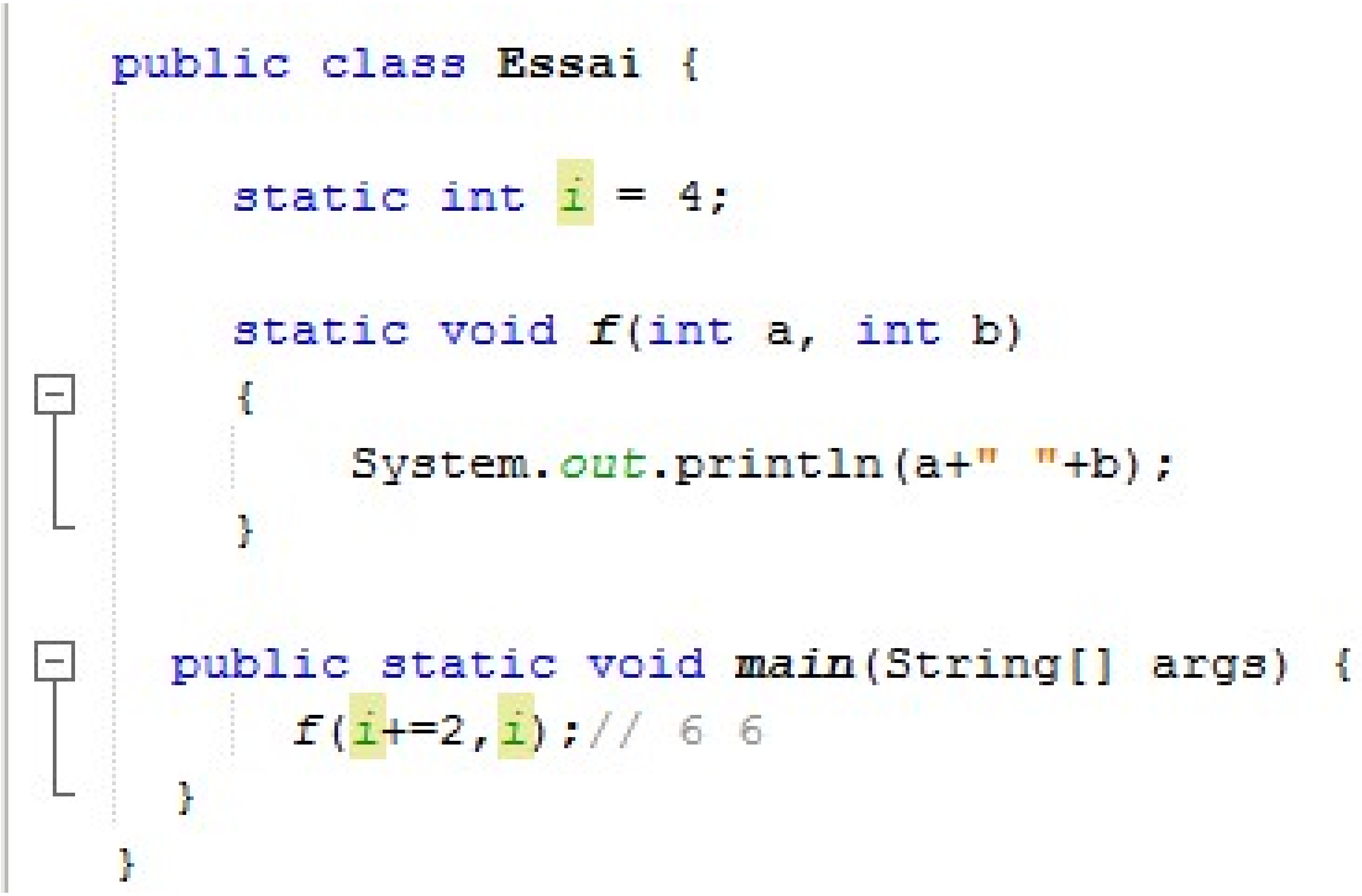
```
    public static void main(String[] args) {
        int[] a = {5, 1, 12, -5, 163};
        Arrays.sort(a);
        for (int i=0;i<a.length;i++)
            System.out.print(a[i]+" ");
    }
}
```



```
    public static void main(String[] args) {
        int[] a = {5, 1, 12, -5, 163};
        Arrays.sort(a);
        System.out.println(Arrays.toString(a));
    }
}
```

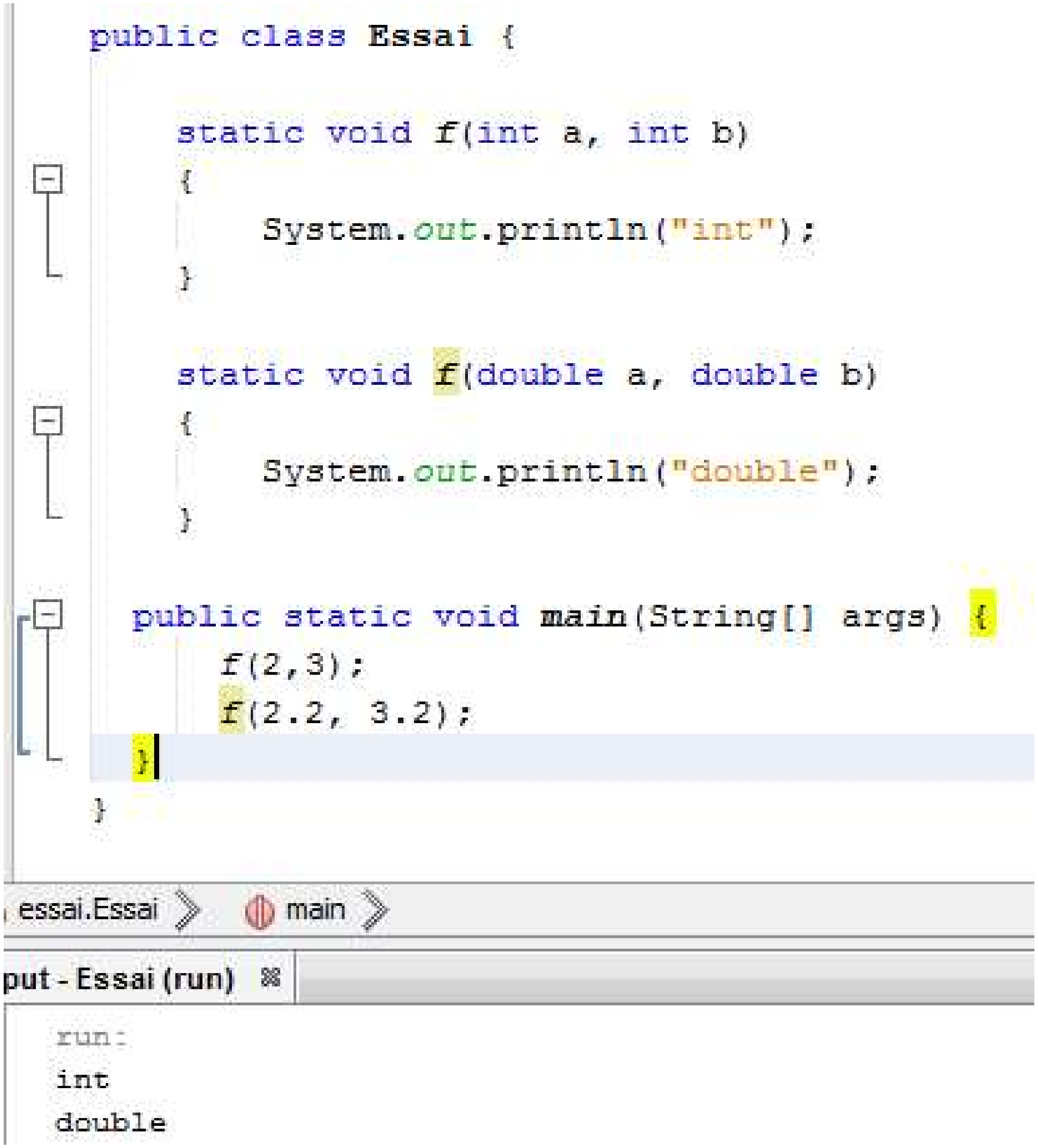
```
run:
[-5, 1, 5, 12, 163]
BUILD SUCCESSFUL (total time: 0 seconds)
```

- L'évaluation des paramètres de gauche à droite



```
public class Essai {

    static int i = 4;

    static void f(int a, int b)
    {
        System.out.println(a+" "+b);
    }

  public static void main(String[] args) {
      f(i+=2,i);// 6 6
  }
}
```

La surcharge des méthodes



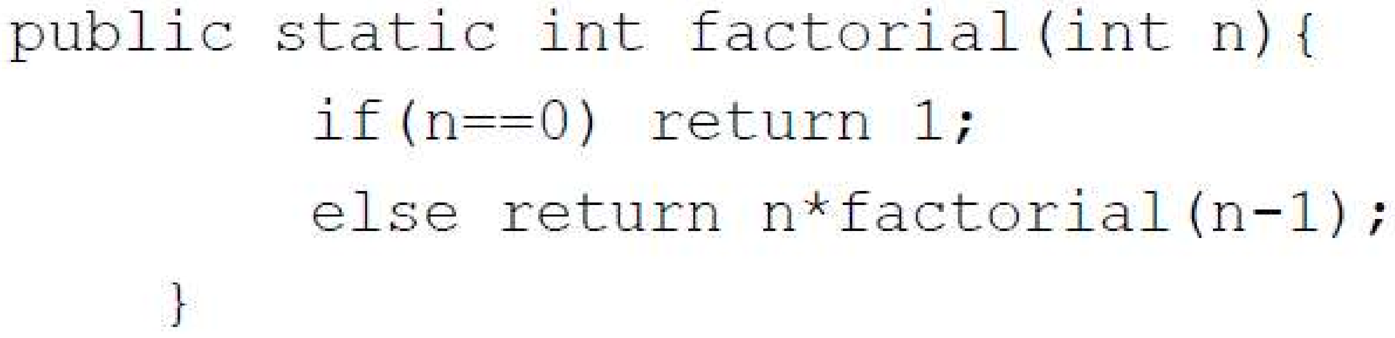
```java
public class Essai {

    static void f(int a, int b)
    {
        System.out.println("int");
    }

    static void f(double a, double b)
    {
        System.out.println("double");
    }

  public static void main(String[] args) {
      f(2,3);
      f(2.2, 3.2);
  }
}
```

```
run:
int
double
```

Recursion

- Une solution d'un problème dépend de petites solutions au même problème
- Fonction récursive = fonction qui appelle lui-même
  - cas de base
  - cas récursives
- Exemple: fonction factorielle

$$\text{fact}(n) = \begin{cases} 1, & n=0 \\ n*\text{fact}(n-1), & n > 0 \end{cases}$$



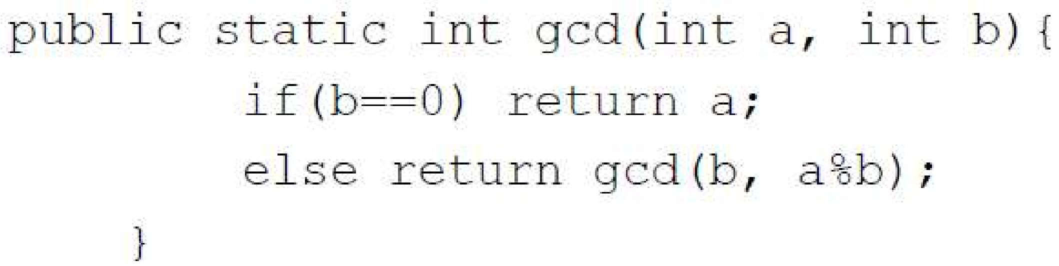
```
public static int factorial(int n){
        if(n==0) return 1;
        else return n*factorial(n-1);
    }
```

Fibonnaci

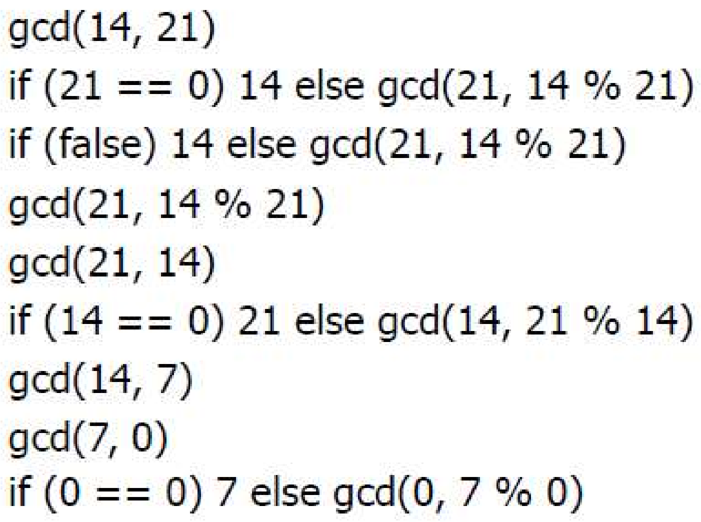
$$\text{fib}(n) = \begin{cases} 0, n=0 \\ 1, n=1 \\ \text{fib}(n-1) + \text{fib}(n-2), n>=2 \end{cases}$$

```
public static int fib(int k) {
        if (k < 2)  return k;
        return fib(k-1) + fib(k-2);
    }
```

Plus grand diviseur commun



```
public static int gcd(int a, int b){
        if(b==0) return a;
        else return gcd(b, a%b);
    }
```



```
gcd(14, 21)
if (21 == 0) 14 else gcd(21, 14 % 21)
if (false) 14 else gcd(21, 14 % 21)
gcd(21, 14 % 21)
gcd(21, 14)
if (14 == 0) 21 else gcd(14, 21 % 14)
gcd(14, 7)
gcd(7, 0)
if (0 == 0) 7 else gcd(0, 7 % 0)
```

https://codippa.com/calculate-gcd-of-two-numbers-in-java/