

# Langages avec des objets : JAVA

Robert Cori

Cours 2 : 29 janvier 2007

**Voir :** `dept-info.labri.fr/ENSEIGNEMENT/poo`

## Vos Travaux Dirigés

- David Auber Groupe 4 : Vendredi 14h -> 16h 50
- Jean-Claude Ville Groupe 2 : Lundi 15h30 -> 18h 30  
Groupe 1 : mardi 9h 30 -> 12h 20
- Anne Vialard Groupe 3 : Vendredi 14h -> 16h 50

## Menu du Jour (cours)

1. Corrigé du Problème
2. Références : la notion de mémoire
3. Les tableaux
4. La classe `String`

## Vu la semaine dernière:

### Langage Java, introduction.

- Généralités: mise en œuvre
- Classes
- Constantes
- Variables et Types
- Affectations
- Structures de contrôle
- Boucles
- Fonctions
- Tableaux : (introduction rapide)
- Chaînes de caractères : (introduction rapide)

## Devoir : Exercice 1

```
public class Prenom {
    public static void main(String[] arg) {
        if (arg[0].compareTo("Aubert")== 0)
            System.out.println("David");
        else if (arg[0].compareTo("Ville")== 0)
            System.out.println("Jean-Claude");
        else if (arg[0].compareTo("Vialard")== 0)
            System.out.println("Anne");
        else if (arg[0].compareTo("Cori")== 0)
            System.out.println("Robert");
    }
}
```

## Devoir : Exercice 1

```
public class Departement {
    public static void main(String[] arg) {
        if (arg[0].compareTo("Bordeaux")== 0) System.out.printl
        else if (arg[0].compareTo("Bergerac")== 0) System.out.p
        else if (arg[0].compareTo("Dax")== 0) System.out.printl
        else if (arg[0].compareTo("Pau")== 0) System.out.printl
    }
}
```

Exception in thread "main"

```
java.lang.ArrayIndexOutOfBoundsException: 0
at Prenom.main(Prenom.java:)
```

## Devoir : Exercice 2

```
class Ex2{

    static int calcul(int n, int x){
        System.out.println(" calcul " + n + " " + x);
        if (n/x == 1 && x != 0) return 1;
        else return 1 + calcul(n, x*x);
    }

    public static void main (String[] args) {
        int m = Integer.parseInt(args[0]);
        int k = Integer.parseInt(args[1]);
        System.out.println(calcul(m, k));
    }
}
```

## Devoir : Exercice 3

```
public class Monnaie{
    static final int[] val = {100, 50, 20, 10, 5, 2, 1};
    static final int n = 7;
```

Ce programme calcule les billets et pieces a rendre pour une

```
void affiche (int[] rep){
    for (i = 0; i <= n ; ++i)
        System.out.println("Nbre de billets de " +
            val[i] + " euros: " + rep[i]);
}
static int[] calcul (int somme){
    int[] r = new int[n];
    m = somme;
    for (int i = 0; (i < n && m != 0) ; ++i)
        while (m >= val[i])
            r[i]++; m = m - val[i];
```



## Devoir : Exercice 3

```
static void main(String[] arg) {  
    int a = parseInt(arg[1]);  
    int t = calcul(a);  
    affiche(t)  
}  
}
```

# Devoir : Exercice 4

## Références

- Case Mémoire
- Variable, valeur
- Affectation
- Passage de paramètres

# Les tableaux

- Pourquoi des tableaux?
- Un tableau est un ensemble de variables de même type
- Déclaration
- Une variable de type tableau est une REFERENCE
- Création `typ`
- Les noms des variables sont `tab[0]`, `tab[1]`, ... `tab[p-1]`
- Le nombre d'éléments du tableau `tab` est donné par `tab.length`
- Affectation
- Egalité
- Paramètre dans une fonction
- Tableaux de taille 0

## Déclaration et initialisation d'un tableau

Pour désigner la variable `alpha` formellement comme un tableau d'entiers

```
int[] alpha;
```

Pour réserver de place en mémoire pour la variable `alpha` comme un tableau composé de  $n$  entiers

```
alpha = new int[n];
```

on peut faire les deux en même temps

## Fonctions sur les tableaux

Une fonction peut avoir pour résultat un tableau :

```
static int[] tabConstant (int a, int n) {  
    int[] res = new int[n];  
        for (int i = 0 ; i < n ; ++i)  
            res[i] = a;  
    return res;  
}
```

## La classe `String`

- Se déclare par:

```
String u;
```

- Création par:

```
u = ``Bonjour tout le monde``;
```

- C'est un **OBJET** sur lequel on peut faire certaines opérations.

## Chaînes de caractères : fonctions utiles

**Méthode** `length()`

```
int l = u.length();
```

Donne le nombre de caractères de la chaîne

**Méthode** `compareTo`

```
u.compareTo(v);
```

Donne pour résultat **0** si les deux chaînes sont égales, négatif si **u** vient avant **v** dans l'ordre lexicographique, positif sinon



## Chaînes de caractères (fin)

- On imprime une chaîne par

```
String u = `` Universite Bordeaux 1``;  
System.out.println(u);
```

- D'ailleurs tous les arguments de `System.out.println` sont transformés automatiquement en chaînes de caractères

- La concaténation se réalise par `+`

```
String x = ``Bonjour``;  
String y = ``les amis``;  
String z = x + y;
```

- L'affectation

```
String x = ``bonjour``;  
String y = x;
```

ne recopie pas “bonjour” mais effectue une référence sur cette chaîne.

## L'opération +

L'opération + désigne à la fois l'addition (entiers, flottants) et la concaténation de chaînes

```
class Plus{  
  
public static void main(String[] arg){  
    String x = " Bonjour ", y = "les amis ";  
    String z = x + y;  
    x = "Au revoir";  
    System.out.println(z);  
    System.out.println(x);  
    int a = 3, b = 5;  
    System.out.println(a + z + b);  
    System.out.println(a + b + z);  
    System.out.println(z + a + b);  
}  
}
```

```
java Plus
```

```
    Bonjour les amis
```

```
    Au revoir
```

```
    3 Bonjour les amis 5
```

```
    8 Bonjour les amis
```

```
    Bonjour les amis 35
```

## Les arguments de main

- C'est la variable déclarée dans `main(String[] arg)`
- Ainsi `arg` est un tableau de chaînes de caractères.
- A l'exécution de `java Nomclasse abc xyz`
- Le nombre d'arguments est `arg.length`
- Le premier argument est `arg[0]` le deuxième `arg[1]` etc ..
- Ces arguments sont de type `String`
- Il faut parfois les transformer en des nombres entiers par `Integer.parseInt`

## Exemple d'entrées avec des arguments pour `main`

```
public class Additionner{
    public static void main (String[] arg) {
        int a = Integer.parseInt (arg[0]);
        int b = Integer.parseInt (arg[1]);
        int res  = a + b;
        System.out.println (c);
    }
}
```