

TD 2 – Passage de paramètres

Concepts informatiques (CI2)

2011–2012

1 Fonctions/programmes paramètres/entrées

Soit le code Java suivant :

```
class A {
    public static void main(String [] args) {
        Scanner s = new Scanner(System.in);
        int a = s.nextInt();
        int b = s.nextInt();
        System.out.println(a+b);
    }
}
class B {
    static int somme(int a,int b) {
        return a+b;
    }
}
class C {
    static int somme() {
        Scanner s = new Scanner(System.in);
        int a = s.nextInt();
        int b = s.nextInt();
        return a+b;
    }
}
```

Indiquez quelles sont les instructions qui correspondent à des appels de fonction, avec quels paramètres, quelle valeur de retour, quelles lignes correspondent à la définition d'une fonction, quels paramètres sont déclarés...

Faites-vous une différence entre les deux fonctions somme ?

2 Échauffement

Quel résultat est affiché par ce programme ?

```
class Exercice {
    public static void ajouterUn(int x) {
        x=x+1;
    }
    public static void ajouterUn(int []tab) {
        tab[0]=tab[0]+1;
    }
    public static void main(String []a) {
        int x=10;
        System.out.println(x);
        ajouterUn(x);
        System.out.println(x);

        int []t=new int[3];
        t[0] = 10;
        System.out.println(t[0]);
        ajouterUn(t);
        System.out.println(t[0]);
    }
}
```

Est-ce différent si on écrit:

```
public static int ajouterUn(int x) {
    x=x+1;
    return x;
}
```

Doit-on changer autre chose, pour que l'effet voulu soit réalisé ?

Faites un schéma du contenu de la mémoire lorsque le point d'exécution est dans la fonction `ajouterInt`, `ajouterTab`.

On rappelle que les éléments d'un tableau unidimensionnel sont stockés en mémoire dans des cases consécutives sous la forme :

t[0]	t[1]	t[t.length-1]
------	------	--------	---------------

Si vous avez déjà écrit quelque chose comme :

```
int []tab=new int[10];
//...
System.out.println(tab);
```

Vous avez obtenu à l'écran [I@10b62c9 (par exemple). Ici [I désigne le type de tab: soit tableau d'entiers. Et 10b62c9 est l'adresse mémoire à laquelle se trouve le tableau. C'est la référence du tableau. La méthode `ajouterUn(int [] tab)` reçoit en argument une copie de la référence du tableau.

3 Tableaux, copie

Écrire une méthode statique qui reçoit un tableau d'entiers en argument, et renvoie un (autre) tableau où tous les éléments ont été augmentés de 1, sans modifier le tableau initial. Écrire une ligne de programme faisant appel à cette fonction.

Écrire une méthode statique qui reçoit en argument un tableau à deux dimensions et renverse les lignes (la première devient la dernière, etc).

4 Objets, passage par référence

On définit le type suivant:

```
class IntRef {
    public int valeur;
    public IntRef(int n) { valeur=n; }
}
```

La fonction de Syracuse est définie, pour un entier $n > 0$ par :

$$f(n) = \begin{cases} n/2 & \text{si } n \text{ est pair,} \\ 3n + 1 & \text{si } n \text{ est impair.} \end{cases}$$

Écrire, en Java, une fonction statique de nom `syracuse` qui reçoit en argument un `IntRef` correspondant à la valeur n et change sa valeur en $f(n)$.

Écrire un programme qui demande à l'utilisateur une valeur initiale n et calcule les valeurs successives $f(n), f(f(n)), \dots, f^p(n), \dots$, jusqu'à obtenir 1. Attention, rien ne garantit que le programme terminera car il n'y a pas de preuve qu'il existe bien un p pour lequel $f^p(n) = 1$ pour tout n . On l'a simplement vérifié expérimentalement jusqu'à 2^{62} .

On crée deux `IntRef` de la manière suivante :

```
IntRef a=new IntRef(10);
IntRef b=new IntRef(3);
syracuse(b);
if (b==a)
    System.out.println("égaux");
else
    System.out.println("pas égaux");
```

Est-ce correct ? Pourquoi ? Comment corriger ?

5 Représenter des tableaux à plusieurs dimensions

Dans certains langages de programmation comme le C, il n'y a pas véritablement de tableau à plusieurs dimensions. On simule alors leur fonctionnement en utilisant des tableaux à une dimension et une fonction permettant de calculer un indice.

- Donner un système simple permettant de représenter un tableau d'entiers de 10 lignes et 5 colonnes à l'aide d'un tableau à une seule dimension. Comment accéder à l'élément de ligne i et de colonne j ?
- Écrire les instructions correspondantes en Java.
- Écrire une fonction de nom `elementAt` et recevant en entrée: un tel tableau et deux entiers i et j . Cette fonction permettra de renvoyer l'élément de coordonnées i, j du tableau vu comme tableau de 10 lignes et 5 colonnes. Pensez à vérifier que le tableau a la taille suffisante...
- Faire la même chose pour un tableau de dimension 3.
- On suppose maintenant qu'un tableau à 2 dimensions est construit avec un tel codage, mais que sa taille dépend de valeurs $T1$ et $T2$ entrées par l'utilisateur (que l'on ne connaît pas quand on écrit le programme). Modifiez, la fonction `elementAt` de sorte qu'elle fonctionne toujours correctement ?

6 Tableau d'objets

Soit la classe `Point`:

```
class Point {
    public double abscisse;
    public double ordonnee;
    public Point(double a, double b) {abscisse = a; ordonnee = b;}
}
```

Écrire une fonction de nom `barycentre`, prenant en entrée un tableau de `Point` et renvoyant leur isobarycentre (centre de gravité).