

## TP n°6

### Les tableaux en *Java*

## 1 Premiers tableaux en Java

Nous désirons créer un programme permettant de manipuler un tableau de réels. Écrivez un programme qui crée un tableau de 3 cases et qui affecte des valeurs constantes à chaque case (les valeurs sont spécifiées directement dans le code). Le contenu du tableau est ensuite affiché.

1. Écrivez le programme.
2. Ajoutez l'instruction suivante à votre programme (si votre tableau ne s'appelle pas `t`, modifiez l'instruction).

```
System.out.println(t);
```

3. Qu'est-ce qui est affiché ?
4. Modifiez le programme pour que la taille du tableau soit saisie par l'utilisateur, ainsi que le contenu de chaque case. Le tableau sera ensuite affiché.
5. À la suite du programme (c'est-à-dire sans modifier les lignes déjà écrites), déterminez le maximum du tableau et affichez-le à l'écran.

## 2 Les notes

Nous souhaitons écrire un programme Java permettant de gérer des notes d'étudiants. Pour cela, nous allons écrire plusieurs fonctions/procédures qui pourront éventuellement être réutilisées dans les prochains exercices et/ou TP.

1. Écrivez une fonction/procédure qui prend un tableau de réels en paramètre et qui le remplit de notes saisies au clavier. Les notes saisies doivent être comprises entre 0 et 20. Le tableau passé en paramètre est déjà alloué.
2. Écrivez une fonction/procédure qui prend un tableau de notes en paramètre et qui l'affiche à l'écran. Le tableau passé en paramètre est alloué et initialisé.
3. Écrivez un main qui demande à l'utilisateur de saisir le nombre de notes puis toutes les notes. Finalement, il affiche les notes saisies à l'écran.
4. Modifiez le programme pour qu'il affiche en plus, la moyenne des notes. Pour cela, nous souhaitons utiliser une fonction/procédure.

Les notes saisies ne sont pas nécessairement sur 20. Nous souhaitons laisser la possibilité à l'utilisateur de choisir (la valeur sera saisie par l'utilisateur).

5. Que doit-on modifier dans le programme ?
6. Modifiez votre programme en conséquence.

### 3 Le tri par sélection

Dans cet exercice, nous souhaitons implémenter l'algorithme de tri par sélection qui permet de trier les valeurs d'un tableau par ordre croissant (ou décroissant). Cet algorithme utilise deux tableaux : `t1` est le tableau à trier et `t2` contiendra à la fin de l'algorithme les éléments de `t1` triés. L'algorithme est le suivant :

- Recherche de l'indice de la case contenant le minimum de `t1`.
- Placer le minimum dans la première case de `t2`.
- Remplacer la valeur de la case de `t1` qui contenait le minimum de `t1` par le maximum de `t1` (cela évitera de réutiliser cette valeur).
- Répéter à partir de l'étape 2 jusqu'à ce que `t2` contienne tous les éléments de `t1`.

Pour réaliser cet algorithme, nous allons utiliser plusieurs fonctions/procédures.

1. Écrivez la fonction/procédure `allocation` qui prend en paramètre un entier `n` et qui retourne un tableau de réels de `n` cases.
2. Que se passe-t-il si la valeur `n` est négative ou nulle ?

Nous souhaitons remplir notre tableau de réels par des valeurs aléatoires contenues dans l'intervalle  $[a; b]$ . Pour obtenir des valeurs aléatoires plus lisibles lors de l'affichage, vous pouvez jouer uniquement sur le transtypage pour obtenir des nombres avec 1 ou 2 décimales maximum. Le code est le suivant :

```
// Remplacez 100 par 10 pour 1 décimale, par 1000 pour 3 décimales...
final double cst = 100;
for(int i = 0; i < t.length; i++) {
    t[i] = Math.random() * 10; // Nombre aléatoire dans [0; 10[
    t[i] = t[i] * cst; // On desire garder seulement 2 décimales
    t[i] = (int)t[i]; // On supprime les décimales restantes
    t[i] = t[i] / cst; // On retourne dans l'intervalle souhaité
}
```

Le même algorithme avec une écriture plus condensée :

```
final double cst = 100;
for(int i = 0; i < t.length; i++)
    t[i] = (double)((int)((Math.random() * 10) * cst)) / cst;
```

3. Écrivez la fonction/procédure `aleatoire` qui retourne un réel aléatoire dans l'intervalle  $[a, b]$  dont les bornes sont passées en paramètre.
4. Comme pourrait-on modifier la fonction/procédure si on souhaitait spécifier la précision en paramètre ?
5. Écrivez la procédure `remplirAleatoire` qui prend en paramètre un tableau alloué et deux réels `a` et `b`. Le tableau est rempli par des valeurs aléatoires dans l'intervalle  $[a, b]$ .
6. Écrivez la procédure `afficher` qui prend un tableau en paramètre et qui l'affiche à l'écran. Les valeurs sont affichées sur une seule ligne, séparées par un espace.
7. Écrivez la fonction/procédure `indiceMinimum` qui retourne l'indice de la case contenant le minimum de toutes les valeurs d'un tableau passé en paramètre.

8. Écrivez la fonction/procédure `maximum` qui calcule le maximum de toutes les valeurs d'un tableau passé en paramètre.
9. Créez la fonction/procédure `triSelection` qui implémente l'algorithme du tri par sélection.
10. Écrivez un programme qui demande à l'utilisateur la taille du tableau et les bornes de l'intervalle  $[a, b]$ . Le tableau est alloué, initialisé puis affiché. Le tableau est ensuite trié puis affiché à l'écran.

## 4 La représentation des binaires



Cet exercice n'est pas obligatoire.

On souhaite réaliser une classe outils `Binaires` contenant un ensemble de fonctions/procédures pour manipuler les binaires. Un binaire est représenté sous la forme d'un tableau d'entiers (faites attention au choix du type).

1. Écrivez une fonction/procédure `entier2binaire(valeur, nbOctets : entier)` qui prend un entier en paramètre et qui calcule sa représentation binaire sur `nbOctets` octets.
2. Écrivez une fonction/procédure `binaire2entier(binaire : tableau d'entiers)` qui prend un binaire en paramètre et qui retourne sa valeur en décimal.
3. Écrivez une fonction/procédure `somme(binaire1, binaire2 : tableau d'entiers)` qui prend en paramètre 2 binaires et qui en calcule la somme.
4. Écrivez une fonction/procédure `complementA2(binaire : tableau d'entiers)` qui calcule le complément à 2 d'un binaire.
5. Écrivez un programme permettant de tester toutes les fonctions/procédures précédentes.