TD de programmation orientée objet en Java
Les tableaux

Exercice 1 : Manipulations basiques des tableaux

Écrire un programme Java nommé `ManipTableaux`, qui permet à l'utilisateur de saisir un tableau d'entiers de taille n. Le programme demandera à l'utilisateur de saisir la taille du tableau puis sollicitera les valeurs à y insérer une par une. Une fois tous les éléments sont saisis, le programme doit :

- Afficher le tableau sous un format lisible
- Calculer et afficher la moyenne et la variance des éléments dans le tableau
- Retrouver le *min* et le *max* des éléments et les afficher

Exercice 2 : Inversion d'un tableau

Écrire un programme Java, nommé `InversionTableau` qui permet à un utilisateur de saisir un tableau contenant n entiers. Le programme se charge alors d'inverser le contenu du tableau (c.à-d. son premier élément devient le dernier et ainsi de suite). L'inversion du contenu du tableau doit être effectuée sur celui-ci, sans utiliser un deuxième tableau. Voici un exemple de l'affichage souhaité :

```
Saisir la taille du tableau : 4
Saisir l'élément no.1 du tableau : 5
Saisir l'élément no.2 du tableau : 10
Saisir l'élément no.3 du tableau : 2
Saisir l'élément no.4 du tableau : 7
Le tableau original est : [5, 10, 2, 7]
Le tableau inversé est : [7, 2, 10, 5]
```

Exercice 3 : Tri par insertion

Écrire un programme Java appelé `TriInsertion` qui permet de trier un tableau d'entiers dans l'ordre décroissant grâce à l'algorithme de tri par insertion [1]. Le programme demande à l'utilisateur de saisir la taille n du tableau et se charge de le remplir de façon aléatoire (en instanciant un objet de type `Random` et en invoquant sa méthode `nextInt`). Le programme doit alors afficher le tableau initial ainsi que son contenu après chaque itération de l'algorithme de tri.

1. cf. `http://fr.wikipedia.org/wiki/Tri_par_insertion`

Exercice 4 : Comptage des éléments d'un tableau

Écrire un programme Java appelé `OccurrenceCount` qui permet de compter le nombre d'occurrences des éléments présents dans un tableau contenant des entiers compris entre 0 et 9. La taille du tableau est à saisir par l'utilisateur. Le programme se chargera de remplir aléatoirement le tableau (toujours en utilisant un objet de type `Random` mais en bien veillant à ce que les valeurs des entiers soient comprises entre 0 et 9). Le programme affichera pour chaque élément son nombre d'occurrences.

Exercice 5 : Suppression des doublons dans un tableau

Écrire un programme Java nommé `SuppressionDoublons` qui permet à l'utilisateur de remplir un tableau d'entiers de taille n (n étant donné par l'utilisateur). Le programme se charge alors de créer un second tableau qui contient les éléments du premier sans leurs doublons (c.à-d. tout entier qui apparait plusieurs fois dans le premier tableau n'apparait qu'une seule fois dans le second). Voici un exemple d'affichage possible (dans lequel la partie qui correspond à la saisie du tableau a été omise) :

```
[...]
Le tableau original est : [3, 5, 7, 2, 3, 4, 5, 8, 6, 9, 9]
Le tableau sans doublons est : [3, 5, 7, 2, 4, 8, 6, 9]
```

Exercice 6 : Coefficients binomiaux

Écrire un programme Java intitulé `BinomialCoefficients` qui permet de calculer et afficher tous les C_n^k [2] pour tout entier inférieur ou égal à un entier donné (saisi par l'utilisateur) en s'appuyant sur la propriété récursive des coefficients binomiaux (formule de Pascal) [3] :

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}, \forall n, k > 0$$

Les coefficients calculés doivent être stockés dans un tableau à deux dimensions `C` en veillant à bien respecter les notations adoptées pour les coefficients binomiaux

2. cf. `http://fr.wikipedia.org/wiki/Coefficient_binomial`
3. cf. la section « Propriété récursive des coefficients binomiaux d'entiers » du lien susmentionné

(autrement dit, la case `C[n][k]` doit contenir la valeur de C_n^k). Les allocations doivent être effectuées au fur et à mesure en fonction du nombre d'éléments à stocker à chaque niveau du tableau. Un affichage possible lorsque l'utilisateur demande le calcul de tous les coefficients binomiaux pour les entiers inférieurs ou égaux à 8 est donné par l'exemple suivant :

```
Saisir un entier : 8
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
[1, 8, 28, 56, 70, 56, 28, 8, 1]
```