Université de Nice – Sophia Antipolis — Deug MASS MIAS SM 1re année
UEF1 : Informatique & Programmation — 2000–2001

TP N° 11

Tableaux à plusieurs dimensions

Ça va mieux en le révisant . . . La notion de tableau à une dimension se généralise à n'importe quelle dimension. Les tableaux à une dimension sont souvent appelés des *vecteurs*, ceux à deux dimensions des *matrices*, vocabulaire qui fait directement référence aux mathématiques.
L'accés a une case donnée d'un tableau se fera toujours en spécifiant chaque indice entre crochets.
Dans le cas unidimensionnel, on vous a présenté les manipulations de base que sont l'ajout et la suppression d'un élément à un tableau. La taille des tableaux n'est pas modifiable en JAVA. On aura donc recours à une variable entière (par exemple `taille`) qui indique le nombre de cases effectivement utilisées d'un tableau, de façon à pouvoir ignorer les autres. La *recherche séquentielle* et la *recherche dichotomique* sont deux algorithmes classiques de recherche d'une valeur dans un tableau.

1 Lecture de tableau à une dimensions

La classe `Console` du paquetage `unsa` contient la méthode `readDoubleArray`(...) qui prend en argument une chaîne de caractères. Cette méthode renvoie un tableau de `double` contenant des valeurs entrées par l'utilisateur.

Exercice 1) Ecrivez une classe `TP11ex1` dans laquelle vous utilisez cette fonction `readDoubleArray`(...) afin d'entrer un tableau de 10 notes, comprises entre 0 et 20.
Vous ferez afficher la moyenne des notes saisies. ⋄

2 Tableaux à deux dimension

Exercice 2) Petit système dynamique Reprenez votre cours n° 9, aux transparents 14-16. L'idée est de transformer le programme donné dans le cours de façon à stocker les 10 premières configurations dans un tableau de dimension 10x10. Modifiez ce programme de sorte à faire calculer puis afficher les 10 premières configutations. ⋄

Exercice 3) Dans une nouvelle classe, créez un tableau d'entier de dimension 7x7. Remplissez sa première ligne d'entiers aléatoires pris dans l'intervalle $[1, 100]$. Faites en sorte que les six lignes suivantes contiennent toutes les permutations circulaires de la première ligne (on décalera d'un cran à chaque fois). ⋄

Exercice 4) Implantez le produit de matrices vu en cours et vérifiez s'il fonctionne sur le jeu d'essai suivant :

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 2 & 1 & 0 \\ 4 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 10 & 5 & 2 \\ 22 & 11 & 4 \end{pmatrix}$$

⋄

3 Approfondissement

Exercice 5) Considérons le vecteur formé par des coordonnées d'une instance de la classe `Point`. Si on multiplie une matrice de dimension 2x2 par ce vecteur, cela revient à transformer ce point dans le plan. Si la matrice est la suivante, on lui fait subir une symétrie.
Engendrez des objets géométriques et faites-leur subir une telle symétrie.

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$\diamond$