Travaux Pratiques n°2 (Java)

SMI S6 (Module : JAVA)

Objectif

- Ce TP a pour objectif d'apprendre à manipuler des Tableaux multidimensionnels avec le langage JAVA.

Les tableaux définissent un nouveau mode de structuration des données. Un tableau est un agrégat de composants, objets élémentaires ou non, de même type et dont l'accès à ses composants se fait par un indice calculé.

Tache N° 1 : utilisation des tableaux de réels pour manipuler des vecteurs.

On cherche à définir une classe pour représenter des vecteurs de la forme $v = [x_1, x_2, x_3, \dots, x_n]$. Les composantes réelles de chaque vecteur seront représentées par un tableau de réels. La dimension du vecteur, c'est-à-dire le nombre d'éléments du tableau, est fixée par le constructeur de la classe.

- Écrivez en JAVA une classe ***Vecteur*** avec un constructeur dont le paramètre est la dimension du vecteur. La dimension sera une caractéristique de chaque objet de type ***Vecteur*** créé.
- Écrivez une méthode appelée norme qui renvoie la norme d'un vecteur. Puis, ajoutez une méthode normalise pour normaliser les composantes du vecteur courant.

On rappelle que la norme d'un vecteur v est égale à :

$$\|v\| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

et que la normalisation d'un vecteur est donnée par :

$$\bar{v} = \frac{v}{\|v\|} = \left[\frac{x_1}{\|v\|}, \frac{x_2}{\|v\|}, \dots, \frac{x_n}{\|v\|}\right]$$

Écrivez une méthode appelée ***produitScalaire*** qui renvoie le produit scalaire du vecteur transmis en paramètre et du vecteur courant. On vérifiera que les deux vecteurs possèdent bien la même dimension. On rappelle que le produit scalaire de deux vecteurs v et v_0 est égal à :

$$v.v' = \sum_{i=1}^{n} x_i \, x'_i$$

Tache N° 2 : utilisation des tableaux à plusieurs dimensions (matrices).

On définit en JAVA la classe Matrice possédant trois attributs : un tableau à deux dimensions qui contiendra les éléments de la matrice, son nombre de lignes et de colonnes :

```
class Matrice {
        // Invariant : this est une matrice (lignes,colonnes)
        private double [][] m;
        public int nbLignes, nbColonnes;
} // fin classe Matrice
```

Ecrivez les méthodes ***symétrique*** et ***produit*** dont les algorithmes sont décrits ci-dessous. La première méthode teste la symétrie de la matrice courante, et la seconde renvoie une matrice (i.e. de type Matrice) produit de la matrice courante et d'une seconde passée en paramètre. Pour que le produit de deux matrices soit valide, vous vérifierez si le nombre de colonnes de la première est égal au nombre de lignes de la seconde.

Matrice symétrique

On désire tester si une matrice carrée est symétrique ou non par rapport à la diagonale principale. On rappelle qu'une matrice carrée m(n, n) est symétrique si :

$$\forall i, j \in [1, n],\ m_{ij} = m_{ji}$$

L'algorithme consiste à parcourir, à l'aide de deux boucles emboîtées, la demi-matrice supérieure (ou inférieure) et vérifier que *m[i, j]* = *m[j, i]*. Notez qu'il est inutile de tester les éléments de la diagonale (i.e. *i* = *j*). Ci-dessous l'algorithme symétrique.

```
Algorithme symétrique
    {teste si une matrice est symétrique ou non}
    variables l, c type [1,n]
              passymétrique type booléen
    l ← 1
    passymétrique ← faux
    répéter
          l ← l+1
          c ← 1
          répéter
                si matrice[l,c] ≠ matrice[c,l] alors
                     passymétrique ← vrai
                sinon
                     c ← c+1
                finsi
      jusqu'à c = l ou passymétrique
          {passymétrique ou ∀i,j∈[1,l], matrice[i,j]=matrice[j,i]}
    jusqu'à l = n ou passymétrique
    {passymétrique ou ∀i,j∈[1,n], matrice[i,j]=matrice[j,i]}
    rendre non passymétrique
```

Produit de matrices

Soient trois matrices *a(m,p), b(p,n) et c(m,n)*, on désire programmer le produit matriciel $c=a\times b$. Rappelons que les éléments de la matrice c sont tels que :

$$\forall i \in [1,m], \forall j \in [1,n],\ c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

Ci-dessous l'algorithme de produit.

```
constantes
   m = ? {nombre lignes de la matrice a}
   n = ? {nombre lignes de la matrice b et
          nombre colonnes de la matrice a}
   p = ? {nombre colonnes de la matrice c}
variables
  a type tableau [[1,m],[1,p]] de réel
  b type tableau [[1,p],[1,n]] de réel
  c type tableau [[1,m],[1,n]] de réel
  somme type réel

  pourtout i de 1 à m faire
      pourtout j de 1 à n faire
         {∀x∈[1,i-1], ∀y∈[1,j-1], c[x,y]=∑_{k=1}^{p} a[x,k]×b[k,y]}
         somme ← 0
         pourtout k de 1 à p faire
               somme ← somme+a[i,k]×b[k,j]
         finpour
         c[i,j] ← somme
      finpour
  finpour
  {∀i∈[1,m], ∀j∈[1,n], c[i,j]=∑_{k=1}^{p} a[i,k]×b[k,j]}
```

Produit de matrice * vecteur

Écrivez une méthode qui renvoie le vecteur résultat du produit de la matrice courante par un vecteur passé en paramètre. On rappelle que le produit d'une matrice a de dimension $m\times n$ et du vecteur v de dimension n est le vecteur v' de dimension m dont les composantes sont définies par :

$$v'_i = \sum_{k=1}^{n} a_{ik} \times v_k$$