

Programmation en Java

Applications 01 : Bases de la programmation en Java

Pour mener à bien cet atelier, chaque étudiant devra créer un projet par exercice en suivant la notation : *Appli01_ExoX_Prenom_Nom*.

Le *X* doit être remplacé le numéro correspondant de l'exercice. *Prenom* et *Nom* doit être remplacé par votre prenom et nom sans les accents.

Exercice 1

Implémentez en Java un algorithme vous permettant de permuter les valeurs de deux variables
Exemple d'exécution:

```
Entrez x: 1
Entrez y: 2
Avant permutation:
x : 1
y : 2
Après permutation:
x : 2
y : 1
```

Exercice 2

Ecrire un programme qui saisit deux entiers et affiche leur produit.

Exercice 3

Ecrire un programme qui demande deux nombres à l'utilisateur, puis qui calcule et affiche la demi-somme de ces nombres.

Exercice 4

Ecrire un programme qui détermine si un entier saisi est pair ou impair.

Exercice 5

Ecrire un programme qui affiche le plus grand de trois entiers saisis.

Exercice 6

Ecrire un programme qui affiche le plus grand et le plus petit d'une suite d'entiers saisis. Les nombres saisis ne sont pas à conserver en mémoire. La suite se termine avec la valeur 0.

Exercice 7

Ecrire un programme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit pas calculer le produit des deux nombres.

Exercice 8

Ecrire un programme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

Exercice 9

Ecrire un programme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : Plus petit ! , et inversement, Plus grand ! si le nombre est inférieur à 10.

Exercice 10

Ecrire un programme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$

NB : On souhaite afficher uniquement le résultat, pas la décomposition du calcul.

Exercice 11

Ecrire un programme qui permet de calculer la somme des n premiers termes de la « série harmonique », c'est-à-dire la somme :

$$1 + 1/2 + 1/3 + 1/4 + + 1/n$$

La valeur de ***n*** sera lue en donnée.

Exercice 12

Ecrire un programme qui permet de calculer le périmètre et la surface d'un cercle de rayon *r*. Intégrer et proposer la déclaration d'une variable constante.

Exercice 13

Ecrire un programme qui permet à la fois de :

- Saisir 3 réels, calculer et afficher leur moyenne.
- Ranger ces 3 réels par ordre croissant

Exercice 14

Ecrire un programme qui demande la note d'un étudiant. Les appréciations suivantes sont données en fonction de la note :

- ✓ « Trop nul » si la note est inférieure à 5
- ✓ « Nul » si la note est inférieure à 10 et supérieure ou égale à 5
- ✓ « pas trop nul » si la note est inférieure à 15 et supérieure ou égale 10
- ✓ « trop top » si la note est inférieure à 20 et supérieure ou égale à 15
- ✓ « T'es un boss » si la note est égale à 20.

Le programme affichera la note et l'appréciation correspondante

Exercice 15

Ecrire un programme :

- Qui calcule le volume d'un pavé droit ;
- Qui affiche ce volume en précisant les dimensions correspondant au pavé droit ;

Exercice 16

Écrivez un programme `Scalaire.java` qui calcule le produit scalaire de deux vecteurs, **implémentés au moyen de tableaux unidimensionnels**. Votre programme devra utiliser (entre autre) les éléments suivants :

- Déclarations dans la méthode `main()`:
 - une variable `nMax` représentant la taille maximale des vecteurs (inutile de lui donner une valeur trop élevée... 10 suffit amplement)
 - deux variables `v1` et `v2`, de type tableau de réels de taille `nMax`.
- Méthode :

- demander à l'utilisateur d'entrer n , la taille effective des vecteurs.
- vérifier que n est compris entre 1 et n_{Max} et demander à l'utilisateur d'entrer à nouveau une valeur tant que ce n'est pas le cas
- demander à l'utilisateur les composantes ($v1_0 \dots v1_{n-1}$, $v2_0 \dots v2_{n-1}$) des vecteurs $v1$ et $v2$.
- calculer le produit scalaire de $v1$ et $v2$.
- afficher le résultat.

Rappel :

Le produit scalaire de a par b est: $a \cdot b = a_1 \cdot b_1 + a_2 \cdot b_2 + \dots + a_n \cdot b_n$

Exemple: $a = (5, 3, -1)$ $b = (2, 1, 2)$ $a \cdot b = 11$

Exercice 17

On cherche ici à écrire un programme `MulMat.java` qui calcule la multiplication de deux matrices (rappel ci-dessous).

Vous utiliserez pour représenter la matrice un tableau de tableaux de double.

Déclarations :

- Dans la méthode `main`, déclarez deux matrices `mat1` et `mat2`.

Traitements :

- Lire depuis le clavier les éléments de chacune des deux matrices (après avoir demandé leurs dimensions).
- Multiplier les deux matrices et stocker le résultat dans une nouvelle matrice `prod`.
- Afficher le contenu de cette nouvelle matrice ligne par ligne.

Méthode :

- Lire depuis le clavier les dimensions `lignes` (nombre de lignes) et `colonnes` (nombre de colonnes) de la première matrice `mat1`
- Lire le contenu de `mat1`.
- De même, lire les dimensions puis le contenu de la seconde matrice `mat2`.
- Vérifier que le nombre de lignes de `mat2` est identique au nombre de colonnes de `mat1`.

Dans le cas contraire, afficher un message d'erreur "Multiplication de matrices impossible !". (rappel: si l'on multiplie deux matrices $M = M1 \cdot M2$, les dimensions de M sont "nombre de lignes de $M1$ " et "nombre de colonnes de $M2$ ", et l'élément $M_{i,j}$ est défini par

- afficher le résultat ligne par ligne.

Exemple d'utilisation:

Saisie de la 1ère matrice :

Nombre de lignes : 2

Nombre de colonnes : 3

M[1,1]=1.0

M[1,2]=2.0

M[1,3]=3.0

M[2,1]=4.0

M[2,2]=5.0

M[2,3]=6.0

Saisie de la 2ème matrice :

Nombre de lignes : 3

Nombre de colonnes : 4

M[1,1]=1.0

M[1,2]=2.0

M[1,3]=3.0

M[1,4]=4.0

M[2,1]=5.0

M[2,2]=6.0

M[2,3]=7.0

M[2,4]=8.0

M[3,1]=9.0

M[3,2]=0.0

M[3,3]=1.0

M[3,4]=2.0

Résultat :

38.0 14.0 20.0 26.0

83.0 38.0 53.0 68.0