

Java Initiation

Exercice 1 : java de base

TP 1.1 : Racine d'un nombre

Ecrire un programme qui calcule la racine carrée de nombres fournis par l'utilisateur.

Ce programme devra afficher une erreur si l'utilisateur fournit des valeurs négatives, et s'arrêter lorsque l'utilisateur entre la valeur 0.

L'exécution du programme devra ressembler aux lignes ci-dessous :

```
Entrez une valeur positive (0 pour terminer) : 2
La racine de 2 est : 1.414214
Entrez une valeur positive (0 pour terminer) : -1
Erreur, la valeur ne peut pas etre negative.
Entrez une valeur positive (0 pour terminer) : 9
La racine de 9 est : 3.000000
Entrez une valeur positive (0 pour terminer) : 0
Fin du programme.
```

Remarque Voir `java.lang.Math.sqrt`

TP 1.2 – Nombre binaire

- Ecrire un programme permettant de :
 - Transformer un nombre binaire vers son équivalent décimale
 - Transformer un nombre décimal en binaire
- Exemple d'exécution demandé :

Voulez-vous convertir :

- 1- un nombre décimal en binaire
- 2- un nombre binaire en décimal

Rentrer votre choix : 2

Rentrer un nombre binaire : 11

Résultat : 11 vos 3 en décimal

Voulez-vous recommencer [oO] ? N

Merci a bientôt

TP 1.3 : Suite

Calculer la somme des n premiers termes de la suite

$$U_n = U_{n-1} + 1 / n \text{ avec } n > 0 \text{ et } U_1 = 1$$

La valeur de n sera fournie par l'utilisateur.

L'exécution du programme devra ressembler aux lignes ci-dessous :

```
Entrez le nombre de terme de la suite à calculer n avec n > 0
(0 pour terminer) : 2
U2 est : 1.5000
Entrez le nombre de terme de la suite à calculer n avec n > 0
(0 pour terminer) : -1
Erreur, la valeur ne peut pas etre negative.
Entrez le nombre de terme de la suite à calculer n avec n > 0
(0 pour terminer) : 3
U2 est : 1.8333
Entrez le nombre de terme de la suite à calculer n avec n > 0
(0 pour terminer) : 0
Fin du programme.
```

Ecrire et tester une méthode de la classe TP qui calcule la somme des éléments d'un tableau de flottants.

La taille du tableau sera fournie par l'utilisateur

```
Entrez la taille du tableau (0 pour terminer) : 3
Entrez valeur 1 : 2
Entrez valeur 2 : 1,5
Entrez valeur 3 : 2,5
La somme du tableau est 6.0
Entrez la taille du tableau (0 pour terminer) : -1
Entrez une taille positive (0 pour terminer) : 1
Entrez valeur 1 : 2
La somme du tableau est 2.0
Entrez la taille du tableau (0 pour terminer) : 0
Fin du programme.
```

TP 1.5 : Tableau

Écrire un programme qui demande à l'utilisateur de taper le contenu d'un tableau de réels de 3 lignes et 3 colonnes et qui affiche ce tableau mais en affichant la moyenne des éléments de chaque ligne, de chaque colonne.

```
Entrez la taille du tableau (0 pour terminer) : 3
Entrez valeur [1,1] : 1
Entrez valeur [1,2] : 2
Entrez valeur [1,3] : 3
Entrez valeur [2,1] : 10
Entrez valeur [2,2] : 20
Entrez valeur [2,3] : 30
Entrez valeur [3,1] : 2
Entrez valeur [3,2] : 4
Entrez valeur [3,3] : 6
```

```
1      2      3      (2,0)
10     20     30     (20,0)
2      4      6      (4,0)
(4,33) (8,66) (13)  (26,0)
Entrez la taille du tableau (0 pour terminer) : 0
Fin du programme.
```

TP 1.6.1 : Méthode d'Archimède 1/2

Créer une méthode Cn de la classe Archimede qui prends :

- » X un double ≥ 0
- » Y un double ≥ 0

La méthode calcule la valeur $2 * x * y / (x + y)$: Utilisation Archimede.Cn(X ,Y)

Tester la Méthode Archimede.Cn

Créer une méthode In de la classe Archimède qui prends :

- » X un double ≥ 0
- » Y un double ≥ 0

La méthode calcule la valeur $\text{sqrt}(x * y)$: Utilisation Archimede.In(X ,Y)

Tester la Méthode Archimede.In

Ecrire la méthode An (et la tester) qui code l'algorithme suivant (en utilisant l'exercices 2.1) N est un entier passer en paramètre:

$X = 4$

$Y = 2 * \text{sqrt}(2)$

Pour $i = 1$ jusqu'à N faire

$X = \text{Archimede.Cn}(X, Y)$

$Y = \text{Archimede.In}(X, Y)$

renvois un tableau contenant X et Y