

EXERCICES Java Série 3: POO

I- Mise en situation

Tu es développeur dans une société et tu dois passer un test en langage Java. A travers une série d'exercices, tu dois comprendre et maîtriser le langage Java pour obtenir la prime salariale.

II- Travail à réaliser

1. Effectuer l'exercice proposé par le professeur.
2. Analyser l'énoncé du point III correspondant au numéro de l'exercice demandé.
3. Réaliser l'exercice.
4. Visualiser le travail.
5. Sauvegarder le document suivant les instructions données.
6. Imprimer le(s) document(s).

III- Enoncés

1. Ex01

Définir une classe **Fibonacci** contenant 2 variables d'instance (**nbAAfficher** (int) et **suite** (String)), 1 *constructeur* sans paramètre alors nbAAfficher vaut 5 et un message explicatif est affiché; 1 *constructeur* à 2 paramètres, les méthodes **accesseurs** et la méthode **toString**.

Prévoir une méthode **privée calculer**(int nb) *réursive* qui renvoie le résultat de la suite de Fibonacci pour un nombre donné et une méthode **lister**(int nbAAfficher) qui ne renvoie rien et qui appelle la méthode calculer() un certain nombre de fois pour garnir la suite qui sera affichée.

Définir une classe **TestFibonacci** sous forme d'*application* qui saisit un nombre au clavier qui détermine le nombre d'éléments de la suite à afficher (premier élément = 0). Si le nombre saisi est négatif ou supérieur à 40 alors on appelle le constructeur sans paramètre sinon celui à deux paramètres.

2. Ex02

Définir une classe **Hanoi** contenant 4 variables d'instance (**nbDisques** (int), **de**, **tmp** et **vers** (String)), 1 *constructeur* sans paramètre alors nbDisques vaut 3 et de= « A », tmp= « B » et vers= « C »; 1 *constructeur* à 1 paramètre alors de= « A », tmp= « B » et vers= « C »; 1 *constructeur* à 3 paramètres (le poteau temporaire est défini automatiquement par rapport à celui de début et celui de fin), les méthodes **accesseurs** et la méthode **toString**.

Prévoir une méthode **privée déplacer**(int nbDisques, String de, String temp, String vers) *réursive* qui effectue et affiche les déplacements et une méthode **effectuerDeplacement**() qui initialise le lancement des déplacements.

Définir une classe **TestHanoi** sous forme d'*application* qui saisit un nombre au clavier qui détermine le nombre de disques à transférer (pour les constructeurs avec paramètre(s)).