

---

**Ecole Supérieure de Management, Télécommunication  
et d'Informatique Oujda**  
**TP : Programmation Orientée Objet en Java**

**Partie 5 :**

**A. Exercice 1 :**

1. Concevez la classe Singleton qui adhère aux critères suivants :

- Uniquement une instance de cette classe doit être présente en mémoire à tout moment.
- Le constructeur ne doit pas être accessible de l'extérieur.
- La classe doit fournir une méthode pour obtenir l'unique instance disponible.

2. Créez un fichier **Main.java** pour démontrer que votre classe Singleton maintient une unique instance en mémoire.

**B. Exercice 2 :**

Créez une classe nommée **CompteurInstances** qui intègre un mécanisme de comptage de ses instances. Les spécifications sont les suivantes :

- Chaque fois qu'une nouvelle instance de **CompteurInstances** est créée, un compteur d'instances doit être incrémenté. Ce compteur doit être commun à toutes les instances de la classe.
- Le constructeur de la classe doit être public pour permettre la création d'instances.
- Incluez une méthode statique **getNombreInstances()** qui retourne le nombre actuel d'instances créées de cette classe.
- Assurez-vous que le compteur d'instances ne peut pas être modifié de l'extérieur, sauf par la création de nouvelles instances de **CompteurInstances**.

➔ Pour démontrer le fonctionnement de votre classe :

- Créez une classe **TestCompteurInstances** contenant une méthode **main**.
- Dans cette méthode, instanciez plusieurs objets de type **CompteurInstances** et utilisez la méthode **getNombreInstances()** pour afficher le nombre d'instances créées.

**C. Exercice 3 :**

1. Écrivez un programme qui inverse l'ordre des éléments d'un tableau d'entiers. Ne pas utiliser de bibliothèque externe pour l'inversion.
2. Écrivez une fonction qui prend en entrée un tableau d'entiers et retourne un tableau de deux éléments, contenant respectivement le minimum et le maximum du tableau d'entrée.
3. Écrivez une fonction qui vérifie si un tableau d'entiers contient un certain élément. La fonction doit retourner **true** si l'élément est présent, **false** sinon.
4. Écrivez une fonction qui effectue une rotation des éléments d'un tableau d'entiers vers la droite de k positions. Par exemple, avec le tableau [1, 2, 3, 4, 5] et k = 2, le tableau après rotation serait [4, 5, 1, 2, 3].

**Méthodes Utiles :**

La classe **Arrays** du package **java.util** en Java offre des méthodes statiques pour faciliter diverses opérations sur les tableaux :

- **sort**: Trie les éléments d'un tableau.
- **binarySearch**: Effectue une recherche binaire (recherche dichotomique) dans un tableau trié pour trouver un élément spécifique.
- **copyOf** et **copyOfRange**: Créent une copie d'un tableau ou d'une partie de celui-ci.
- **equals**: Compare deux tableaux pour vérifier leur égalité.
- **fill**: Remplit un tableau avec une valeur donnée.
- **toString**: Retourne une représentation sous forme de chaîne de caractères du contenu d'un tableau.