

PROGRAMMATION ORIENTÉE OBJET (POO)

Application au langage Java

Travaux pratiques

POO

Série N° 3

Tableaux & fonctions de tri

Java

Youssef EL ALLIOUI

y.elalloui@usms.ma

Exercice 1. Classe Clavier

Ecrire la classe *Clavier* qui fournies les méthodes statiques permettant de lire ; sur une ligne console ; une information de type *int*, *float*, *double* ou *String*. Cette classe contiendra alors :

- Une méthode statique *lireString* () qui lit une chaine de caractères au clavier
- Des méthodes statiques *lireInt* (), *lireDouble* () et *lireFloat* () pour convertir les données lus en *int*, *float* et *double*.

Créer une classe *Main* qui possèdera une méthode *main* () pour réaliser des appels à la classe *Clavier*.

Exercice 2. Tableau des carrés des n premiers nombres impairs

Écrire un programme qui crée un tableau comportant les valeurs des carrés des n premiers nombres impairs, la valeur de n étant lue au clavier et qui en affiche les valeurs sous la forme suivante :

```
Combien de valeurs : 5
1 a pour carre 1
3 a pour carre 9
5 a pour carre 25
7 a pour carre 49
9 a pour carre 81
```

Pour se faire :

- Créer une classe *UtilTab*.
- Créer une méthode statique *estImpair* () qui permet de savoir si un nombre, fourni comme paramètre, est impair ou non
- Créer une méthode statique *premiersNombresImpairs* () qui permet de renvoyer un tableau des n (fourni come paramètre) premiers nombres impairs.
- Créer une méthode statique *carresNombres* () qui renvoie les carrés des nombres (fournis sous forme de tableau comme paramètre) sous la forme suivante : $\{\{2,4\}, \{3,9\}, \dots\}$

- Créer une méthode statique *afficherTableauBidimensionnel()* pour l’affichage d’un tableau bidimensionnel sous la forme :

```
1 a pour carre 1
3 a pour carre 9
...
```

De plus, La classe *UtilTab* disposera des méthodes statiques suivantes :

- *somme()* qui fournit la somme des valeurs d’un tableau de réels (double) de taille quelconque,
- *incrémenter()* qui incrémente d’une valeur donnée toutes les valeurs d’un tableau de *double*.
- *décrémenter()* qui décrémente d’une valeur donnée toutes les valeurs d’un tableau *double*.

Créer une classe *Main* qui possèdera une méthode *main()* pour réaliser des tests sur les classes créées.

Exercice 3. Application de tri

On veut créer une application qui permet de trier des tableaux d’objets en particulier des objets d’une classe nommée *Réel* :

- Ecrire une interface *ITrie* qui contient :
 - Une méthode *trier()* qui renvoie un tableau d’objets, fourni en paramètre, trié.
 - Une méthode *trier()* qui renvoie un tableau d’objets, fourni en paramètre, trié en selon un ordre à spécifier.
 - Deux constantes, de type *int*, statiques *ASC* et *DESC* pour spécifier l’ordre de tri.
- Créer une classe *TrieParInsertion* qui implémente l’interface *Itrie*.
- Créer une classe *TrieABulle* qui implémente l’interface *Itrie*.
- Créer une classe *TrieRapide* qui implémente l’interface *Itrie*.
- Créer une classe *Reel* qui contient
 - Un nombre *double*.
 - Des constructeurs.
 - Des *getters* et des *setters*.
- Une classe *Reels* qui contient :
 - Un tableau de taille *n* (la taille est fournie lors de l’instanciation)
 - Un objet de type *Itrie* (fourni en paramètre)
 - Des constructeurs (avec et sans paramètres)
 - Une méthode *ajouterReel()* pour ajouter un *double* au tableau
 - Une méthode *supprimerReel()* pour supprimer un *double* du tableau d’indice *i*
 - Une méthode *copierTableau()* qui crée une copie du tableau
 - Une méthode statique *calculerMoyenneTableau()* qui calcule et renvoie la moyenne d’un tableau fourni comme paramètre
 - Une méthode statique *elementMaxTableau()* qui renvoie le plus grand *double* d’un tableau fourni comme paramètre
 - Une méthode statique *elementMinTableau()* qui renvoie le plus petit *double* d’un tableau fourni comme paramètre
 - Une méthode statique *trier()* qui renvoie un tableau trié selon un ordre (à spécifier) et une technique de trie (objet de type *Itrie*) d’un tableau fourni comme paramètre.

Créer une classe de *Main* qui possèdera une méthode *main()* pour réaliser des tests sur les classes créées.