

**TP3 :Tableaux****Enseignants : Madhi Kamel, Majdoub Manel****Groupes :** Prepa A02**Objectifs :** Manipulation des tableaux.**Exercice 1**

On veut enregistrer dans un objet de la classe ArrayList les notes obtenues par un élève à une matière.

Le programme devra comporter les opérations suivantes, chacune réalisée par une méthode prenant en paramètre un objet de la classe ArrayList contenant les notes : ajouter une nouvelle note,afficher toutes les notes, calculer la moyenne.

Le programme principal devra réaliser un test des méthodes avec les notes 12, 14, 9, un calcul de ma moyenne, puis ajout de la note 13 et nouveau calcul de la moyenne et affichage des notes.

Une nouvelle méthode permettra de changer une note du tableau. Cette méthode peut être utile en cas d'erreur dans la saisie ou dans la correction de la copie.

**Exercice 2**

1. Définir une classe Pile permettant de représenter une pile d'entiers stockés sous forme d'un tableau. Les méthodes à prévoir sont :
  - Constructeur de la pile qui reçoit en argument la taille de la pile.
  - void empile (int e) : met l'entier e à la fin de la pile.
  - void depile () : enlève le dernier élément de la pile.
  - void affiche () : affiche le contenu de la pile si la pile contient des éléments, sinon affiche le message : « La pile est vide ».
2. Ecrire une classe TestPile permettant de tester la création d'une pile de 10 entiers ainsi que les différentes méthodes créées.

**Exercice 3 : Etagères de livres (tableaux d'objets)****Partie 1**

Créez une classe Livre ne possédant que des attributs privés sachant qu'un Livre est caractérisé par : son titre, son auteur, le nombre de page qui le constitue et le prix.

- (a) Doter cette classe d'un (des) constructeur (s) adéquat.
- (b) Doter cette classe des méthodes pour récupérer ou accéder aux différents attributs.

**Partie 2**

Créez une classe Etagere pour représenter une étagère qui peut contenir un certain nombre de livres (fixe pour chaque étagère). Vous utiliserez pour cela un tableau.

- (a) Créez un constructeur de la classe Etagère. Ajoutez des méthodes permettant de :
- (b) Donner le nombre de livres que peut contenir l'étagère, et le nombre de livres qu'elle contient.
- (c) Ajouter des livres (boolean ajouter(Livre)). Vous ajouterez les livres "à la fin" de l'étagère. Il devra être impossible d'ajouter des livres dans une étagère pleine.
- (d) Récupérer un livre dont on donne la position sur l'étagère (le livre reste sur l'étagère, on récupère simplement une référence sur le livre). La méthode renverra une instance de Livre. La position du premier livre d'une étagère devra être 1 (et pas 0, bien que le livre soit rangé dans la première position du tableau, qui est d'indice 0). La signature de la méthode sera Livre getLivre(int).
- (e) Chercher sur l'étagère un livre repéré par son titre et son auteur. La méthode renverra la position du livre dans l'étagère (ou 0 si le livre n'y est pas). Le profil de la méthode sera int chercher (String, String). S'il y a plusieurs livres avec le même titre et le même auteur, la méthode renvoie celui qui a le plus petit indice.
- (f) Avoir une fonctionnalité semblable à la précédente, mais la méthode renvoie un tableau de positions s'il y a plusieurs livres qui ont ce titre et cet auteur. On aimerait appeler cette méthode "chercher" ; est-ce possible ? Le tableau aura pour taille le nombre de livres trouvés (0 si aucun livre n'a été trouvé).
- (g) Ecrivez aussi 2 méthodes pour rechercher tous les livres d'un auteur, ou tous les livres qui ont un certain titre. Cette fois-ci, les méthodes renvoient un tableau de livres.
- (h) Enlever des livres. Vous "tasserez" les livres vers le début quand vous enlèverez des livres. Vous écrirez 2 méthodes de même nom pour enlever un livre : une méthode qui repèrera le livre par sa position (1 pour le premier livre) dans l'étagère (de profil Livre enleverLivre(int)), une méthode qui repèrera le livre par son auteur et son titre, et qui utilisera la méthode chercherLivre (de profil Livre enleverLivre(String, String)).  
Les 2 méthodes renverront le livre supprimé (ou null si le livre n'a pas été trouvé).
- (i) S'il y a beaucoup de livres, cet algorithme ("tasser" les livres, et donc faire un grand nombre de décalages) n'est pas bon pour enlever un livre. Réécrivez les méthodes enleverLivre pour placer le dernier livre à la place du livre enlevé ; vous nommerez les nouvelles versions "enlever".
- (j) Dans la méthode main(), vous créerez des livres, 2 étagères et ajouterez les livres dans les étagères. Vous chercherez un des livres dans une étagère ; s'il y est, vous ferez afficher son nombre de pages. Vous rechercherez tous les livres d'un auteur et les ferez afficher. Vous supprimerez un des livres d'une étagère.

#### Conseils

- Lire l'énoncé en entier avant de répondre aux différentes questions
- Commencez par écrire le constructeur, la méthode pour ajouter un livre et testez. N'écrivez les autres méthodes que lorsque l'ajout de livre fonctionne correctement.