

Laboratoire 03 – Tableaux et enregistrements

OBJECTIFS :

- Vous exercer à manipuler des tableaux et des enregistrements.
- Continuer à bien modulariser vos applications en regroupant des méthodes « similaires » dans des modules utilitaires (`UtilitaireTabReels`) réutilisables.
- Vous familiariser avec quelques méthodes d'une classe standard de Java : `Arrays`.
- Continuer à vous exercer à respecter les bonnes pratiques de programmation vues au cours et le style Java.
- Continuer à vous exercer à faire des sous-programmes et optimiser votre code (éviter la répétition, favoriser la réutilisation).

EXERCICE 1

Vous devez écrire un module utilitaire nommé `UtilitaireTabReels` qui contient des méthodes manipulant des tableaux de nombres réels (`double`).

1. Ajoutez dans votre projet la classe `UtilitaireTabReels` (fournie avec cet énoncé), qui est un module ne contenant, pour le moment, que la coquille des méthodes que vous devez implémenter. Lisez bien la Javadoc d'une méthode avant de la coder, pour savoir exactement ce qu'elle doit faire.
2. Implémenter les méthodes une par une, en commençant par la première, dans l'ordre où elles sont définies dans le module. **Chaque fois que vous avez terminé l'implémentation d'une méthode, testez-là, et déboguez-là avant de passer à la suivante.**

Pour tester vos méthodes, ajoutez à votre projet la classe `TestsUtilitaireTabReels` (fournie avec cet énoncé) qui contient déjà des tests pour les deux premières méthodes que vous devez implémenter dans la classe `UtilitaireTabReels`. Pour tester les autres méthodes, vous devrez y ajouter vos propres tests.

EXERCICE 2

1. Créez une nouvelle classe dans votre projet pour définir un enregistrement nommé `Cours` qui contient les champs suivants :
 - `sigle` (`String`)
 - `groupe` (`int`)
 - `tabNotes` (un tableau de type `double`) auquel on ajoutera les notes (/100) pour ce cours.
 - `nbrNotes` (`int`) qui conserve le nombre de notes significatives dans `tabNotes`.
2. Créez une autre classe nommée `ExercicesLabo3` et ajoutez-y les méthodes suivantes, en les codant / testant à mesure qu'elles sont codées.

```
/**
 * Initialise les champs d'un cours avec les parametres donnees.
 *
 * - Le champ tabNotes doit etre initialise avec un tableau vide.
 * - Le champ nbrNotes doit etre initialise a 0.
 *
 * @param cours le cours qu'on veut initialiser
 * @param sigle le sigle a assigner a ce cours
 * @param groupe le groupe a assigner a ce cours
 */
public static void initCours (Cours cours, String sigle, int groupe)
```

INF111 – Programmation orientée-objet

```
/**
 * Affiche les champs du cours donne en parametre, dans le format explique
 * ci-dessous.
 *
 * Exemples d'affichage :
 *
 * Pour le cours INF111, groupe 03, sans aucune note, la methode
 * devrait afficher :
 *     INF111-03 (aucune note)
 *
 * Pour le cours INF111, groupe 10, avec un tabNotes = [78.3, 90, 87.5, ...]
 * et avec nbrNotes = 3, la methode devrait afficher :
 *     INF111-10 [78.3, 90, 87.5]
 *
 * @param cours le cours dont on veut afficher les champs
 */
public static void afficherCours (Cours cours)

/**
 * Ajoute la note donnee, au tableau des notes du cours donne, a la
 * position donnee.
 *
 * Si la position donnee n'est pas valide, l'ajout n'est pas effectue.
 *
 * ANT: on suppose que cours.tabCours n'est pas null.
 *
 * NOTE : UTILISER UtilitaireTabReels
 *
 * @param cours le cours auquel on veut ajouter une note
 * @param note la note a ajouter a ce cours
 * @param position la position a laquelle ajouter la note dans tabNotes.
 */
public static void ajouterNote(Cours cours, double note, int position)

/**
 * Supprime la note a la position pos donnee, dans le tableau des notes du
 * cours donne.
 *
 * Si la position donnee n'est pas valide, la suppression n'est pas
 * effectuee.
 *
 * ANT: on suppose que le cours.tabCours n'est pas null.
 *
 * NOTE : UTILISER UtilitaireTabReels et n'oubliez pas de decrementer
 * le nombre de notes.
 *
 * @param cours le cours duquel on veut supprimer une note
 * @param position la position de la note a supprimer dans cours.tabNotes.
 */
public static void supprimerNote(Cours cours, int position)
```

3. Dans votre classe `ExercicesLabo3`, écrivez une méthode `main` et ajoutez-y les instructions pour accomplir les tâches suivantes (en réutilisant votre code au possible) :
- Créez le cours `INF111-03` (disons `c1`)
 - Affichez `c1` et vérifiez que les informations sont correctes : `INF111-03 (aucune note)`
 - Ajoutez les notes suivantes à `c1` :
 - 67.4 à l'indice 0
 - 73 à l'indice 0

- 88.5 à l'indice 2
 - 77 à l'indice 1
 - 75.9 à l'indice 3
 - 86.7 à l'indice 1
 - 91 à l'indice 7 //indice invalide, ne doit pas être ajouté
- d. Affichez c1 et vérifiez que les informations sont correctes :
INF111-03 [73.0, 86.7, 77.0, 67.4, 75.9, 88.5]
- e. Supprimez les notes de c1 aux indices suivants (DANS L'ORDRE) :
0, 2, 3, 4 (4 étant invalide, ne supprime pas)
- f. Affichez c1 et vérifiez que les informations sont correctes :
INF111-03 [86.7, 77.0, 75.9]
- g. Ajoutez la note 66 à c1, à l'indice 1.
- h. Affichez c1 et vérifiez que les informations sont correctes :
INF111-03 [86.7, 66.0, 77.0, 75.9]
- i. Créez 2 autres cours (c2 et c3), et initialisez-les : c2 : INF254-10 et c3 : INF155-05
- j. Ajoutez les notes suivantes à c2 : 55.6 (à l'indice 0) et 34.9 (à l'indice 1).
- k. Créer un tableau de type `Cours`, de longueur 5.
- l. Ajoutez-y c1 à l'indice 0, c2 à l'indice 2, et c3 à l'indice 4.
- m. Écrivez une **boucle** pour afficher tous les cours dans le tableau, puis la moyenne des notes de chaque cours. Votre code devrait afficher :

```
INF111-03 [86.7, 66.0, 77.0, 75.9]
  Moyenne : 76.4
INF254-10 [55.6, 34.9]
  Moyenne : 45.25
INF155-05 (aucune note)
  Moyenne : 0.0
```

BON TRAVAIL !