

TP N°3 – R2.01

Partie 1 : La classe Promotion

- Récupérer le code Java de la classe `Etudiant` que vous avez écrit et testé la semaine 5.
- Ecrire le code de la classe `Promotion` vu en TD, en n'oubliant pas la javadoc.
- Ecrire une autre classe qui définit des scénarios d'exécution pour la classe `Promotion`, on l'appellera `ScenarioPromotion` :
 - parmi les scénarios on étudiera le passage par référence d'un tableau d'objet en créant un tableau d'objets de type `Etudiant`, que l'on passera en paramètre du constructeur de la classe `Promotion`. Ensuite on modifie ce tableau dans `ScenarioPromotion` et on observe l'état du tableau dans la classe `Promotion`.
- Générer la javadoc pour les classes `Promotion` et `Etudiant`.

Partie 2 : Gestion de tableau partiellement rempli

Nous voulons assouplir l'utilisation de la classe `Promotion` en permettant l'inscription d'étudiants au fur et à mesure.

Nous n'allons plus passer en paramètre un tableau déjà rempli avec des étudiants, mais nous allons créer un tableau vide et le remplir au fur et à mesure. Il faut donc gérer un tableau partiellement rempli.

Cela va impacter le code de plusieurs méthodes. Il faut donc faire une copie de la classe `Promotion` que vous pouvez appeler `Promotion2`.

1. Ajouter un attribut **`nbInscrits`** à la classe `Promotion` qui permettra de conserver le nombre d'étudiants inscrits dans la promotion et qui sera l'indice de remplissage du tableau (le nombre réel d'éléments).
2. Modifier le premier constructeur pour la classe `Promotion` (`Promotion2`) pour ajouter l'initialisation de l'attribut **`nbInscrits`** avec la taille de `listeEtudiants`. Dans ce cas on ne pourra plus inscrire de nouveaux étudiants.
3. Définir un nouveau constructeur pour la classe `Promotion` qui ne prend pas de tableau d'étudiant en paramètre, mais simplement le nom de la promotion et une taille maximum pour le tableau.
4. Ajouter une méthode nommée **`inscrire`** à la classe `Promotion` qui prend un étudiant en paramètre et qui permet d'ajouter/inscrire un nouvel étudiant en vérifiant que le tableau ne déborde pas et en gérant correctement l'indice de remplissage.
5. Modifier les tests de fin dans les méthodes.
6. Ecrire une classe `ScenarioPromotion2`, inscrire des étudiants et vérifier le fonctionnement des méthodes, vérifier également que la première version de `Promotion` fonctionne encore.

Travail à rendre :

Mettre dans une archive vos dossiers contenant tout le code source et la javadoc générée.

Déposer l'archive sur la zone moodle

Remarque :

Tous les étudiants doivent au moins déposer la partie 1.

Il est recommandé d'essayer de faire la partie 2.