Associations et collections

1 Promotions

Vous avez déjà modélisé et implémenté une classe Etudiant. Nous allons étudier une classe Promotion, qui utilisera la classe Etudiant.

Question 1. Une promotion est un ensemble d'étudiants, pour une année donnée. Nous représentons cette relation entre la classe `Etudiant` et la classe `Promotion` par une association (voir Figure 1).

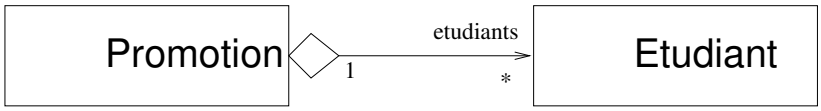

FIGURE 1 – Classes Etudiant et Promotion

a- Mettez en place la classe `Promotion` et l'association entre `Promotion` et `Etudiant`. Pour cela, on placera dans la classe `Promotion` une collection d'étudiants sous la forme d'une `ArrayList`. Cette collection aura une visibilité privée et ses accesseurs ne devront pas permettre sa modification directe (retour d'une collection non modifiable comme vu en cours). On veillera à bien mettre dans la classe Promotion un attribut représentant l'année, les accesseurs associés, et deux constructeurs : un constructeur sans paramètres, et un constructeur prenant une année en paramètre. Les constructeurs initialiseront l'année et créeront la collection. La documentation de la classe ArrayList de Java est disponible via la documentation de l'API Java :
`https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html`

b- Écrire deux méthodes publiques manipulant la collection : une première retournant le ième étudiant de la collection, et une seconde retournant le nombre d'étudiants de la promotion.

c- Mettez en place la classe TestPromotion (avec un `main`) qui vous permet de tester votre classe `Promotion`. Vous commentez tout le contenu en le mettant dans un environnement `/* */` et vous décommentez au fur et à mesure que vous écrivez vos méthodes pour les tester.

```
public class TestPromotion {

    public static void main(String[] args) {
        /*Pour une forme de constructeur Etudiant(String nom, int anneeNaiss,
        CodePays codePays,
        double note1, double note2, double note3)*/

        Etudiant e1 = new Etudiant("Marie", 1998, CodePays.etuFrançais,18, 18, 18);
        System.out.println(e1);

        Etudiant e2 = new Etudiant("Jeanne", 1998, CodePays.etuFrançais,14, 9, 16);
        System.out.println(e2);

        Etudiant e3 = new Etudiant("Sylvie", 1998, CodePays.etuFrançais,7, 9, 10);
```

```
        System.out.println(e3);

        Etudiant e4 = new Etudiant("Esther", 1998, CodePays.etuFrançais,7, 9, 5);
        System.out.println(e4);

        Etudiant e5 = new Etudiant("Astrid", 1998, CodePays.etuFrançais,10, 10, 0);
        System.out.println(e5);

        Etudiant e6 = new Etudiant("Mohamed", 1998, CodePays.etuEtrangerNonFrancophone,18, 18, 1
        System.out.println(e5);

        Etudiant e7 = new Etudiant("Bjorg", 1998, CodePays.etuEtrangerNonFrancophone,18, 18, 18)
        System.out.println(e5);

        Promotion p = new Promotion("groupe PEIP",2022);

        /* les instructions suivantes ne doivent pas pouvoir être exécutées
        p.getEtudiants().add(e2);
        p.getEtudiants().add(e2);
        p.getEtudiants().add(e3);*/

        // tester les cas limites (ici promotion vide)

        System.out.println("\n---PROMO VIDE----\n"+p.getEtudiants());
        System.out.println("\n---PROMO VIDE----\n"+p.moyenneGénérale());
        System.out.println("\n---PROMO VIDE----\n"+p.moyenneGénéralev1());
        System.out.println("\n---PROMO VIDE----\n"+p.recherche("Astrid"));
        System.out.println("\n---PROMO VIDE----\n"+p.recherche("astrid"));
        System.out.println("\n---PROMO VIDE Admis----\n"+p.admis());
        System.out.println("\n---FIN PROMO VIDE----\n");

        // tester le cas général
        p.inscrire(e1);p.inscrire(e2);
        p.inscrire(e2);// ne sera pas réinscrit (pas de doublon)
        p.inscrire(e3);p.inscrire(e4);
        p.inscrire(e5);p.inscrire(e6);
        p.inscrire(e7);
        System.out.println("\n-------\n"+p.getEtudiants());
        System.out.println("\n-------\n"+p.moyenneGénérale());
        System.out.println("\n-------\n"+p.recherche("Astrid"));
        System.out.println("\n-------\n"+p.recherche("astrid"));
        System.out.println("\n---Admis----\n"+p.admis());
        System.out.println("\n---Etu Et. non francophone ----\n"
                        +p.etuEtrangerNonFranco());
        System.out.println("\n---Majors ----\n"+p.majors());
    }
}
```

Question 2. Complétez la classe `Promotion` avec les méthodes suivantes (on veillera à bien tester chacune des méthodes au fur et à mesure, grâce à la classe `TestPromotion`) :

— une méthode `inscrire` qui permet d'inscrire un étudiant dans la promotion ;
— une méthode `moyenneGenerale` qui retourne la moyenne générale de la promotion ;
— une méthode `afficheResultats` qui affiche une ligne pour chaque étudiant (correspondant au résultat de la méthode `ligneResultat`) ;
— une méthode `recherche` qui permet de retrouver un éudiant d'après son nom. On suppose qu'il n'y a pas d'homonymes ;
— une méthode `admis` qui retourne l'ensemble des étudiants admis ;
— une méthode `etuEtrangerNonFranco` qui retourne l'ensemble des étudiants non francophones, elle pourrait être utilisée pour connaître les étudiants susceptibles de suivre des cours d'apprentissage en français ;
— une méthode `majors` qui retourne les étudiants dont la moyenne est la plus élevée.

2 Transport du courrier

Nous revenons dans cette partie sur les objets postaux introduits lors des TD précédents. Nous allons maintenant nous intéresser à la description des sacs postaux.
Un *sac postal* peut contenir un certain nombre d'objets postaux, et dispose d'une capacité maximale. Cette capacité est de $0,5m^3$ pour un sac ordinaire. On peut fabriquer des sacs d'une autre capacité, sur demande précisant la capacité voulue.
On doit pouvoir ajouter un objet dans un sac, s'il y rentre. On veut connaître le volume occupé par un sac (compter forfaitairement $5dm^3$ pour la toile du sac), et sa valeur de remboursement en cas de perte. Enfin, on désire aussi pouvoir remplir un nouveau sac avec tous les objets de même code postal extraits d'un autre.

Question 3. Complétez le diagramme de classes déjà réalisé pour y intégrer la notion de sac postal.

Question 4. Ecrivez et testez la classe Java correspondante.