

# Mini-application OO en Java

## Exercice 1

a) On décrit un étudiant par ses nom, prénom, date de naissance, adresse mail, adresse postale. Implémentez la classe *Etudiant*, certains attributs étant des chaînes de caractères et d'autres des dates.

Pour représenter les dates on utilisera la classe `LocalDate` du package `java.time`. Vous pouvez consulter la documentation Java en ligne :

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>

Voici un exemple de création d'une date :

```
import java.time.LocalDate;

public class ExempleDate {
    public static void main(String[] args) {
        LocalDate maDate = LocalDate.of(2024, Month.MARCH, 29);
    }
}
```

b) Proposez également un constructeur pour un *Etudiant*.

Pensez maintenant à utiliser les facilités de votre IDE pour générer les méthodes standard, puis à les adapter à vos besoins ensuite. Vous pouvez aussi utiliser les raccourcis clavier, consultez la liste en utilisant `Ctrl+J`.

c) Proposez la méthode `String toString()` qui renvoie un texte présentant les informations à propos d'un *Etudiant*. Souvenez-vous que cette méthode n'affiche rien, elle produit simplement une chaîne de caractères, qui pourra être utilisée à tout endroit du code où une telle chaîne est attendue.

d) Ajoutez la fonction `void setNom(...)` permettant de changer le nom d'un *Etudiant*.

e) Vérifiez l'implémentation des précédentes méthodes dans une classe *GestionEtudiants* contenant la méthode principale `main(...)`.

f) Dans le `main`, construisez un deuxième *Etudiant* avec les mêmes données que le premier. Comparez ces objets avec l'opérateur `==`. Que constatez-vous ?

g) Modifiez votre code pour faire en sorte que le deuxième objet fasse référence au premier. En invoquant la méthode `setNom(...)` sur le premier objet, vérifiez que la modification se répercute bien sur le second.

h) Dans la classe *GestionEtudiants*, ajoutez une fonction visant à remplacer le contenu d'un premier objet par celui d'un second :

`static void copier(Etudiant e1, Etudiant e2)`. Constatez.

- i) Implémentez une classe *Departement* avec pour attributs une spécialité, une adresse et le tableau des étudiants inscrits géré sous forme de liste : vous choisirez la classe *ArrayList* du package *java.util*. Consultez la documentation officielle de cette classe pour en connaître les méthodes les plus utiles.
- j) A la construction d'un objet de cette classe, on renseignera la spécialité et l'adresse. Définissez la méthode *toString()* dans *Departement* retournant un texte qui liste l'ensemble des étudiants inscrits.
- k) Ajoutez dans cette classe la méthode *void inscrire(...)* : elle prend en paramètre un étudiant et l'ajoute aux inscrits.
- l) Ajoutez la méthode *void desinscrire(...)* qui enlève un étudiant de la liste des inscrits.
- m) Dans la classe *GestionEtudiants*, construisez maintenant deux nouveaux étudiants, avec des noms différents de ceux déjà créés. Construisez aussi un département et inscrivez-y les étudiants dans l'ordre suivant : 1<sup>er</sup>, 3<sup>ème</sup>, 2<sup>nd</sup>, 4<sup>ème</sup>, les numéros correspondant à l'ordre de construction. Désinscrivez ensuite le 3<sup>ème</sup> du département. Que constatez-vous ?

## Exercice 2

Commencez par copier l'ensemble des classes du package *exo1* dans le package *exo2* en utilisant les outils de l'IDE : sélectionnez toutes les classes ensemble puis clic droit Refactor → Copy Classes

- a) On souhaite maintenant gérer les notes obtenues par les étudiants dans différentes matières. Une *Matiere* est définie par un intitulé et un coefficient (valeur réelle qui servira dans le calcul d'une moyenne). Implémentez la classe correspondante, ainsi que son constructeur. Ajoutez dans *Matiere* un accesseur pour le coefficient.
- b) Une *Note* est définie par une matière et par un nombre réel (la valeur de la note). Déclarez la classe correspondante et ajoutez des accesseurs pour chacun des attributs.
- c) Ajoutez à la classe *Etudiant* la liste de ses notes, ainsi qu'une méthode *noter(...)* ayant pour paramètres une matière et une valeur réelle (la note de l'étudiant dans cette matière) : cette méthode ajoute une *Note* à la liste des notes de l'étudiant.
- d) Ajoutez à la classe *Etudiant* une méthode *double calculerMoyenne()* qui calcule et retourne la moyenne pondérée des notes de l'étudiant : attention, à ce stade, vous ne devez pas modifier le code des autres classes.
- e) Vérifiez que votre programme fonctionne bien depuis la classe *GestionEtudiants*.

- f) On souhaite ensuite calculer la moyenne de toute la promotion. Ajoutez à la classe *Departement* la méthode *double* *getMoyennePromo()* qui retourne cette moyenne.
- g) Qu'observe-t-on entre les fonctions *calculerMoyenne()* de la classe *Etudiant* et *getMoyennePromo()* de la classe *Departement* ? Quelles modifications imaginer afin d'éviter ce défaut ?