

TP N°2 – R2.01

Soit la classe `Etudiant` dont on donne le diagramme de classes de conception UML ci-dessous. L'objectif est de :

- i) La développer en Java puis
- ii) De « jouer » un peu avec cette classe en montrant l'usage a minima de toutes ses méthodes en cas nominal sur plusieurs instances d'étudiant (**on ne demande pas ici un test unitaire au sens strict du terme**).

Le code que vous développerez devra **impérativement respecter le principe de la programmation défensive** :

- Vérifier la validité des paramètres passés à chacune des méthodes, alerter si nécessaire et permettre le fonctionnement d'une méthode même dans un mode dégradé.
- Garantir la maîtrise totale pour un étudiant des données qu'il gère en réalisant, lorsque nécessaire, des copies défensives.

La classe `Etudiant` sera également doté d'une Javadoc.

Votre classe de jeu comportera uniquement une méthode `main` dans lequel vous disposerez l'intégralité de votre code en lui intégrant des affichages clairs expliquant à chaque étape ce que vous testez. Elle portera le nom de `TestEtudiant`.

Description de la classe

Voici une description de la classe à développer, documenter en Javadoc et tester unitairement.

Etudiant
<pre>-nom : String -bulletin : double[][] -MATIERES : String[] -COEFFICIENTS : double[] +Etudiant(nom : String, matieres : String[], coeff : double[], nbNotes : int) +setNom(nom : String) +getNom() : String +getNbMatiere() : int +getUneNote(iMatiere : int, i : int) : double -initialisation() +moyenneMatiere(iMatiere : int) : double +moyenneGenerale() : double +meilleureNote() : String +toString() : String</pre>

La classe `Etudiant` possède comme attributs :

- un **nom** représenté par une chaîne de caractères
- un tableau **bulletin** à deux dimensions pour stocker les notes de chaque matière.
- Un tableau `MATIERES` qui est un attribut d'instance et qui contient les intitulés des matières.
- Un tableau `COEFFICIENTS` qui est un attribut d'instance et qui contient les coefficients des

matières.

Pour qu'il y ait une correspondance entre les coefficients, les matières et le bulletin on doit garantir que : MATIERES, COEFFICIENTS et la première dimension de **bulletin** aient la même taille.

Coefficient	Matière/Note			
3	Math	10	12	9
2	Chimie	15	14	18
1	Histoire	8	10	11
2	Philo	13	15	14
3	Français	14	6	7

Vous devez développer et tester unitairement a minima et dans l'ordre indiqué les méthodes qui suivent :

- Le **constructeur** qui recevra en paramètres le nom de l'étudiant, le tableau de matières, le tableau des coefficients et le nombre des notes par matières (toutes les matières ont le même nombre de notes, ainsi dans l'exemple ci-dessus trois notes).
- La méthode privée **initialisation()** qui a pour but de remplir le bulletin d'un étudiant avec des valeurs aléatoires réelles entre 0 et 20. On utilisera pour cela `Math.random()` qui fournit une valeur aléatoire comprise entre 0 inclus et 1 exclus.
- Une méthode d'accès à l'attribut nom : **getNom()** qui retourne la chaîne de caractères représentant le nom de l'étudiant.
- Une méthode de modification de l'attribut nom : **setNom(String)** qui prend un chaîne de caractère en paramètre qui est la nouvelle valeur de l'attribut nom.
- Une méthode **getNbMatiere()** qui retourne le nombre de matières
- Une méthode **getUneNote(int, int)** qui prend en paramètre l'indice d'une matière et l'indice d'une note et retourne la note correspondante.
- Une méthode **toString()** qui redéfinit la méthode héritée par défaut de la classe `Object`. Cette méthode retourne une chaîne de caractères représentant les données d'un étudiant (son nom et les notes obtenues par matières) que l'on pourra afficher plus tard lors d'un `println` prenant en paramètre une variable de type `Etudiant`.
- La méthode **moyenneMatiere(int)** prend en paramètre l'indice d'une matière et calcule la moyenne des notes pour la matière et retourne la valeur obtenue. Il faut vérifier la validité du paramètre afin d'éviter une division par 0 en calculant la moyenne. Toutes les notes ont le même coefficient au sein d'une matière.
- Une méthode **moyenneGenerale()** qui calcule la moyenne de toutes les moyennes par matière en appliquant les coefficients de chaque matière, puis retourne la valeur obtenue.
- Une méthode **meilleureNote()** qui retourne une chaîne de caractères contenant la matière dans laquelle a été obtenue la meilleur note.

Bonus : les plus rapides uniquement peuvent commencer à écrire un test unitaire complet (chaque méthode est alors testée dans une méthode `static` à part, dans les cas nominaux, limites et hors-limites). Mais ne vous lancez dans cette voie que si vous avez du temps pour cela !

Travail à rendre :

Une **archive zip** nommé avec votre nom et le numéro de TP contenant :

- votre répertoire **src** avec vos deux classes `Etudiant` et `TestEtudiant`
- et votre répertoire **doc** qui contient la javadoc générée pour la classe `Etudiant`.