

Exercice 2.2

Rappel : Pour préserver le code écrit dans l'exercice précédent, copiez l'ensemble des classes du package `dz.Biskra.Info.exo1` dans le package `dz.Biskra.Info.exo2` en utilisant les outils de *refactoring* de l'IDE. Pour ce faire : sélectionnez les classes **simultanément** → clic droit → *Refactor* → *Copy Classes*

On souhaite étoffer le modèle objet conçu auparavant en y incluant les aspects pédagogiques du département. Pour cela on vous demande de gérer les *matières* et les *notes*.

1. Une *Matiere* est définie par un intitulé, un coefficient (une valeur réelle qui pourra servir pour le calcul d'une moyenne). Créez la classe correspondante avec un constructeur adéquat. Ajoutez dans *Matiere* une méthode accesseur `getCoefficient()` qui retourne le coefficient.
2. Une *Note* est définie par une matière et par un nombre réel (qui représente la valeur de la note). Déclarez la classe correspondante et ajoutez des méthodes accesseurs pour chacun des attributs.
3. Ajoutez à la classe *Etudiant* un attribut correspondant à la liste de ses notes et une méthode `noter(...)` qui prend en paramètre une matière et une valeur réelle, crée un objet de type *Note* et l'ajoute à la liste des notes de l'étudiant.
4. Sans toucher au code des autres classes, ajoutez à la classe *Etudiant* une méthode `double calculerMoyenne()` qui retourne la moyenne pondérée des notes de l'étudiant. Pensez à ajouter des tests unitaires pour vérifier le bon fonctionnement de cette fonction. Pour créer une classe de tests unitaires, placez-vous dans la classe que vous souhaitez tester et :
 - appuyez sur Alt+Insert (ou bien faites un clic droit sur le nom de la classe → *Generate*)
 - Choisissez *Test...*
 - dans l'onglet *Testing library* vous choisirez l'option *JUnit 5 (ou JUnit4)*
 - donnez un nom approprié à votre classe de tests unitaires (par ex. *EtudiantTest*) et cliquez sur *Ok*.

La classe de tests générée sera automatiquement placée dans le même **package** que la classe testée et dans le répertoire correspondant aux tests. Pour écrire vos tests, vous pouvez vous inspirer des exemples vus dans le document de Test sur le site (https://bb17f732-27b0-45aa-957a-56ffb1c74327.filesusr.com/ugd/bf605f_5a8b9ea6eb324f56836bd963f78ab3eb.pdf).

Vous devez ajouter au moins les tests suivants :

- `test_calcul_moyenne_retourne_zero_Quand_Pas_De_Note()`
- `test_calcul_moyenne_retourne_valeur_note_quand_une_seule_note()`
- `test_calcul_moyenne_retourne_valeur_note_pondérée_quand_une_seule_note_avec_coefficient()`
- `test_calcul_moyenne_retourne_moyenne_quand_plusieurs_notes_avec_différents_coefficients()`

Indication : Il est assez difficile de comparer des nombres réels, car dans presque tous les langages de programmation, ils sont représentés en virgule flottante et par conséquent, ils sont *approximés* (plus de documentation pour Java sur le [site d'Oracle](https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html#assertEquals(double, double, double))). Une façon simple de comparer des réels est de les comparer à un epsilon près :

`assertEquals(double expected, double actual, double delta)`

5. Vérifiez que votre programme fonctionne bien en l'exécutant depuis la classe principale.

6. Maintenant, on souhaite calculer la moyenne de toute la promo. Ajoutez dans la classe `Departement` la méthode `double getMoyennePromo()` qui retourne la moyenne de toute la promo et des tests unitaires appropriés dans la classe correspondante.
7. Qu'observez-vous lorsque vous comparez les fonctions `calculerMoyenne()` de la classe `Etudiant` et `getMoyennePromo()` de la classe `Departement` ? Sans changer votre programme, discutez avec l'enseignant quelles seraient les modifications à apporter au programme afin d'éviter cette duplication de code.
8. Construisez le diagramme de classes UML correspondant au programme *Java* que vous avez écrit (vous pouvez vous inspirer des exemples vus [en cours](#)). Vous pouvez utiliser [PlantUML](#) qui est un outil interactif en ligne libre et open-source. La syntaxe est très simple à apprendre pour faire des jolis diagrammes.