# Mini Projet : Calculateur de moyennes/crédits

**Remise et consultation des projets :**
La remise et la consultation du travail est prévue avant les vacances de l'hiver.
**Attention :**
Toute ressemblance entre deux projets fera diviser la note par deux.
Tout étudiant qui ne saura pas expliquer son code lors de la consultation aura une note de zéro.

# I. Sujet

L'objectif du projet est la réalisation d'un programme en java qui permettra de calculer les résultats d'un étudiant dans un semestre à partir de ses notes.

- Le programme reçoit en entrée :
  - La liste des unités d'enseignement du semestre et les matières de chaque unité ;
  - Les notes des étudiants dans chaque matière (notes des examens, des TDs et des TPs).
- En sortie, le programme affiche les résultats de tous les étudiants (Moyenne du semestre et crédit acquis), et permettra à l'utilisateur d'afficher les résultats détaillés (notes, moyenne, et crédit acquis de chaque matière et de chaque unité d'enseignement).

Pour réaliser le programme, on envisage l'implémentation d'un ensemble de classes (`Etudiant`, `UniteEnseigement`, `Matiere`, `MatiereCours`, `MatiereCoursTD`, `MatiereCoursTDTP`), et une interface (`Evaluable`) :

## 1. L'interface `Evaluable`

L'interface `Evaluable` déclare 2 méthodes abstraites `moyenne()` et `creditAcquis()`.

## 2. La classe `Matiere`

- Une matière est caractérisée par :
  - L'intitulé (chaine de caractères);
  - Un crédit (entier) ;
  - Un coefficient (entier);
  - une note de l'examen (réel).

- La classe matière a trois classes filles : `MatiereCours` (Matière avec cours uniquement), `MatiereCoursTD` (Matière avec cours et TD), et `MatiereCoursTDTP` (Matière avec cours, TD, et TP) :
  - Une matière avec cours et TD se caractérise par une note de TD (réel), et une constante `coefTD=0.33`.
  - Une matière avec cours, TD, et TP se caractérise par une note de TD et une note TP (réels), et deux constantes `coefTD=0.2`, et `coefTP=0.2`.
- La classe Matière implémente l'interface `Evaluable`.
  - La moyenne d'une matière avec cours uniquement est égale à la note de l'examen.
  - La moyenne d'une matière avec cour et TD= noteExamen*(1-coefTD)+ note TD*coefTD.
  - La moyenne d'une matière avec cour TD et TP= noteExamen*(1-coefTD-coefTP)+ noteTD*coefTD+ noteTP*coefTP.
  - Le crédit acquis d'une matière égal à son crédit si la moyenne est >=10, et à 0 sinon.

- La classe `Matiere` et chacune de ses classes filles implémentent deux constructeurs : un constructeur qui initialise tous les attributs, et un constructeur qui initialise uniquement les attributs : `intitule`, `credit`, et `coeficient`.

- La classe `Matiere` et chacune de ses classes filles implémentent les accesseurs de lecture et de modification pour chaque attribut.
- La classe `Matiere` et chacune de ses classes filles implémentent une méthode `toString()` qui renvoie une chaine de caractères de la forme :

```
Matiere: Programmation orientee objet
 Credit :5
 Coeficient :3
 Note de l'examen : 10
 [Note TD : 10]
 [Note TP : 10]
 Moyenne: 10
 Credit acquis: 5
```

## 3. La classe `UniteEnseigement`

- Une unité d'enseignement est caractérisée par :
  - Un code (chaine de caractères);
  - Un type (fondamentale, méthodologique, découverte, ou transversale);
  - Une liste de matière (Un objet de la classe `ArrayList`).
- La classe **`UniteEnseigement`** implémente un constructeur qui initialise ses attributs. L'initialisation de la liste des matières se limite à la création d'un objet de la classe `ArrayList` (Les matières seront rajoutées plus tard).
- La classe **`UniteEnseigement`** implémente les accesseurs de lecture et de modification pour tous les attributs.
- La classe **`UniteEnseigement`** déclare les méthodes :
  - `credit()` (retourne la somme des crédits de ses matières) ;
  - `coefficient()` (retourne la somme des coefficients de ses matières) ;

- La classe **`UniteEnseigement`** déclare une méthode `toString()` qui renvoie une chaine de caractères de la forme :

```
============================================================
Unite d'enseignement: UEF212 Credit: 13 Coeficient: 8
 Moyenne: 11.69
 Credit acquis: 13
 ---------------------Liste des matieres--------------------
 Matiere: Programmation orientee objet
 Credit :5
...
------------------------------------------------------------
 Matiere: Systemes d'information
 Credit :4
...
```

- La classe **`UniteEnseigement`** implémente l'interface `Evaluable` :

  - La moyenne d'un UE (de $n$ matières) $= \frac{(\text{moyenne matière 1} * \text{coefficient 1}) + \ldots + (\text{moyenne matière n} * \text{coefficient n})}{\text{Somme des coeficient des n matieres}}$

  - Le crédit acquis d'une UE égal à la somme des crédits de ses matières si sa moyenne >=10, et à la somme des crédits acquis de ses matières sinon.

## 4. La classe `Etudiant`

- Un étudiant se caractérise par :
  - Une filière (Informatique, Mathématique, ou MI);
  - Une année (entier)
  - Un semestre (entier) ;
  - Une liste des unités d'enseignement (Une liste de la classe `ArrayList` ).

- La classe **`Etudiant`** implémente un constructeur qui initialise tous les attributs.

- La classe **`Etudiant`** implémente les accesseurs de lecture et de modification pour tous les attributs.
- La classe **`Etudiant`** l'interface **`Evaluable :`**

  - La moyenne d'un semestre (de $n$ UE) $= \frac{(\text{moyenne UE 1} * \text{coefficient 1}) + \ldots + (\text{moyenne UE n} * \text{coefficient n})}{\text{Somme des coeficient des UE}}$

  - Le crédit acquis d'un semestre égal à la somme des crédits des UE si la moyenne du semestre >=10, et à la somme des crédits des UE sinon.

- La classe **`Etudiant`** déclare une méthode `toString()` qui renvoie un chaine de caractères de la forme :

```
---------------------------
 Numero:1
 Prenom:Idir
 Nom: Mohammed
 Annee: 2
 Semestre: 3
 Moyenne: 11.50
 Credit acquis: 30
```

1. La classe **`Etudiant`** déclare une méthode `toString(boolean compact)` qui retourne un chaine de caractère selon l'un des formats suivants :
   - Si `compact` est égal à `true,` la méthode retourne une chaine semblable à celle de la méthode `toString()`
   - Si `compact` est égal à `false,` la méthode renvoie la chaine précédentes, ainsi que les notes, les moyennes et les crédits acquis par l'étudiant de chaque matière et de chaque UE.

# II. Travail à faire

1. Implémenter les classes et l'interface décrites précédemment
2. Déclarer une classe `MainProgram`, qui ne possède pas d'attributs et possède une unique méthode `main()`qui permettra à l'utilisateur de :
   - Saisir une liste d'UE, ainsi que les matières de chaque unité.
   - Saisir les informations d'un étudiant et ses notes, puis, afficher ses résultats du semestre (moyennes et crédits acquis), ainsi que ses résultats détaillés (moyennes et crédits acquis de chaque UE et de chaque matière).

## Annexe

***ArrayList***

La réalisation du travail nécessite la manipulation de listes de la classe `ArrayList` du package `java.util`. Une description détaillée de cette classe est disponible sur l'adresse :

https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html

Parmi les méthodes de la classe `ArrayList` :

**`add()`** permet d'ajouter un élément ;
**`get(int i)`** retourne l'élément à l'indice i;
**`remove(int i)`** supprimer l'élément l'indice *i*;
**`size()`** retourne le nombre d'éléments dans la liste.
**`clear():`** pour vider la liste.

***Scanner***

La lecture des informations saisies au clavier peut être faite à l'aide de la classe `Scanner` du package `java.util.` Un bon tutoriel sur la lecture des entrées clavier à l'aide de la classe `Scanner` est disponible sur l'adresse :https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/20615-lire-les-entrees-clavier

| | **Maître** | **Coefficient** | **Cour** | **TD** | **TP** |
|---|---|---|---|---|---|
| **UE fondamentale Code : UEF211 Crédit :15 Coefficients :7** | Architecture des ordinateurs | 2 | 5 | 5 | 5 |
| | Algorithmique et structures de données | 3 | 6 | 6 | 6 |
| | Logique mathématique | 2 | 4 | 4 | 4 |
| **UE fondamentale Code : UEF212 Crédit :13 Coefficients :8** | Programmation orientée objet | 3 | 5 | 5 | 5 |
| | Systèmes d'information | 3 | 4 | 4 | 4 |
| | Théorie des langages | 2 | 4 | 4 | 4 |
| **UE méthodologique Code : UEM21 Crédit :2 Coefficients :1** | Langue étrangère 2 | 1 | 2 | 2 | 2 |