

Java : Révision N°1**Exercice 1 : Gestion de scolarité (durée : 35 min)**

Le système de scolarité se compose de trois parties :

Gestion des filières

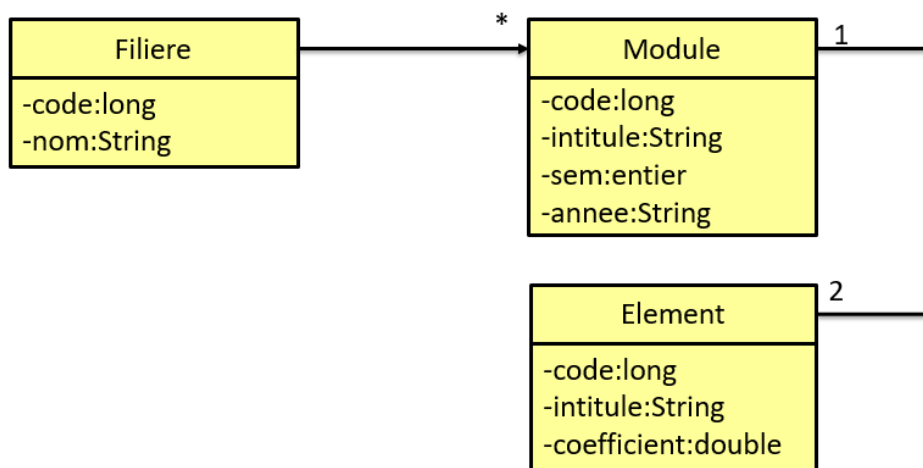
Gestion des inscriptions

Gestion des notes

Dans cette étape, nous allons réaliser les deux parties : gestion des filières et gestion des notes

1. Gestion des filières (10 min)

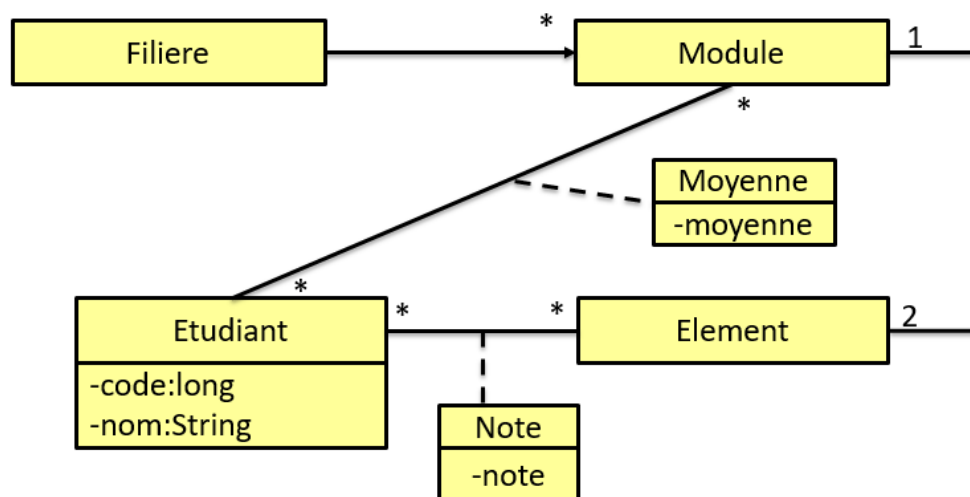
Pour la gestion des filières, on a proposé le diagramme de classes suivant :

**Questions**

- Implémenter le diagramme de classes
- Créer une filière GLSID ou BDCC
- Créer 3 modules pour le même semestre 3 et les associer à la filière créée
- Pour chaque module, créer deux éléments

2. Gestion des notes (25 min)

Pour la gestion des notes, on a proposé le diagramme de classes suivant :



- Implémenter le diagramme de classes (la suite du diagramme précédent)

- Créer 5 étudiants
- Pour chaque élément, saisir les notes des 5 étudiants
- Calculer la moyenne pour chaque module de chaque étudiant
- Afficher la liste des notes de tous les étudiants pour un module donnée sous la forme suivante :

Module 1: UML & Genie du logiciel

aa aa 12

bb bb 13

cc cc 12

dd dd 14

- Afficher le relevé de notes d'un étudiant dont le code est connu sous la forme suivante :

Module 1: UML & Génie du Logiciel

UML 14

GL 14

Moyenne: 14

Module 2: POO en Java

Java1 14

Java avancé 16

Moyenne: 15

Module 3: Communication

Français & TEC 16

Anglais 16

Moyenne: 16

Moyenne semestre 1: 15

Exercice 2 : comptabilisation d'occurrences des caractères à l'aide d'une Map<K,V> (durée 20min)

Soit la chaîne de caractère suivante String s="***+*+++++/////***-----"

Comptabiliser le nombre d'occurrence de chaque caractère **en utilisant une seule boucle à l'aide d'une Map<K,V>**

Une fois ce traitement terminé, le résultat doit être affiché sous la forme suivante :

*** 7**

+ 8

/ 5

- 6

NB. Cet exercice doit se faire sur machine pour consulter les fonctionnalités offertes par JDK (String et Map en particulier)

Exercice 3 : durée (12 min)

```

class S {
    private static S s=null;
    private static int nb=0;
    private String data="une data";
    private S(){
        nb++;
    }
    public static int getNb(){
        return nb;
    }
    public String getData() {
        return data;
    }
    public static S getInstance(){
        if(s==null) s=new S();
        return s;
    }
    public void setData(String data) {
        this.data = data;
    }
}

```

```

class A {
    public void f(){
        S s=S.getInstance();
        System.out.println("ref s dans f: "+s);
        // référence de l'objet créé sous la forme @5678
    }
}

class B {
    public void g(){
        S s=S.getInstance();
        System.out.println("ref s dans g: "+s);
        // référence de l'objet créé sous la forme @1234
    }
}

```

```

public class Program {
    public static void main(String[] args) {
        S s=S.getInstance();
        A a =new A();
        B b=new B();
        System.out.println("ref s dans main: "+s);
        // référence de l'objet créé sous la forme @223344
        a.f();
        b.g();
        System.out.println(S.getNb()); // S représente le nom de la classe
    }
}

```

Donner le résultat du programme ci-dessus si possible
 Si non, indiquer les problèmes qui peuvent avoir lieu.