

Interrogation n°1

La production finale devra être rendue sous la forme du code. Le plus simple est de *zipper* le répertoire du code source, de le renommer de votre nom (ex. : `yunes.zip` si votre nom de famille est `yunes`) et de l'envoyer par mail à `Jean-Baptiste.Yunes@u-paris.fr`. Le rendu final devra contenir un `main` dans une classe `Test` permettant de tester toutes les fonctionnalités implémentées....

1. Créer une classe `Note` avec deux attributs : un pour représenter l'intitulé du cours et un autre pour contenir la valeur de la note (on ira jusqu'au $\frac{1}{2}$ point). L'intitulé du cours ne peut être changé, mais la note oui (une plainte d'un étudiant peut mener jusqu'à recorriger sa copie et réévaluer sa note). La construction d'un objet `Note` doit utiliser deux valeurs pour les attributs. Un affichage par défaut devrait permettre d'afficher correctement une `Note`. Un exemple possible serait le suivant :

```
Note n = new Note("Java", 17);
System.out.println(n);
n.setNote(18);
System.out.println(n);
```

qui pourrait conduire à un affichage comme :

```
[é Java:17]
[Java:18]
```

2. Créer une classe `Étudiant` avec comme attributs : son nom, son prénom, sa date de naissance et sa date d'inscription. Aucun de ces attributs ne pourra être modifié une fois l'étudiant créé, sauf son prénom. Tous ces attributs doivent être correctement initialisés à la construction et l'affichage d'un étudiant doit produire un résultat agréable. Un exemple d'usage de la classe pourrait être le suivant :

```
Etudiant e = new Etudiant("Emilio", "Juñes", 18, 3, 1934);
System.out.println(e);
e.setPrenom("Juan Bautista");
System.out.println(e);
```

qui pourrait produire un affichage comme :

```
Juñes, Emilio, 18 mars 1934, 13 mai 2022.
Juñes, Juan Bautista, 18 mars 1934, 13 mai 2022.
```

3. Modifier la classe `Étudiant` de sorte qu'on puisse à chaque instance associer autant de `Notes` que souhaité. L'affichage d'un étudiant devra être modifié en conséquence afin que l'on puisse voir apparaître toutes ses notes. Par exemple de la façon suivante :

```
Etudiant e = new Etudiant("Santiago", "Juñes", 18, 3, 1934);
System.out.println(e);
Note n = new Note("Java", 2);
e.add(n);
System.out.println(e);
```

qui pourrait produire :

```
Juñes, Santiago, 18 mars 1934, 13 mai 2022, actuellement sans note.
Juñes, Santiago, 18 mars 1934, 13 mai 2022, Notes : [Java, 2].
```

4. Modifier la classe `Étudiant` de sorte que l'on puisse obtenir la moyenne des notes de l'étudiant *via* une méthode `getMoyenne`. Par exemple, le code suivant :

```
Etudiant e = new Etudiant("Santiago", "Juñes", 18, 3, 1934);
e.add(new Note("Java", 2));
e.add(new Note("Tango", 16));
System.out.println(e);
```

pourrait produire le résultat suivant :

```
Juñes , Santiago , 18 mars 1934 , 13 mai 2022 , Moyenne: 9 , Notes : [Java ,2] , [Tango ,16].
```

5. Créer une classe `Cursus` avec un seul attribut non modifiable correspondant à l'intitulé. On pourra ajouter des autant d'étudiants que l'on souhaite dans un cursus. Pour un cursus donné on pourra obtenir la liste des étudiants diplômés (moyenne ≥ 10) et ceux recalés. Un code comme celui-ci pourrait être employé :

```
Etudiant jby = new Etudiant("Juan Bautista", "Juñes", 18, 3, 1934);
Etudiant cd = new Etudiant("Cristóbal", "Darmangua", 19, 8, 1933);
Etudiant jml = new Etudiant("Juan Miguel", "Leri", 22, 2, 1935);
jby.add(new Note("Java", 15));
cd.add(new Note("C#", 3));
jml.add(new Note("C", 1));
Cursus c = new Cursus("PISE");
c.add(jby);
c.add(cd);
c.add(jml);
System.out.println(c);
System.out.println("Sont reçus: "+c.getRecus());
System.out.println("Sont recalés: "+c.getRecales());
```

qui pourrait produire quelque chose comme :

```
PISE:
Juñes , Juan Bautista , 18 mars 1934 , 13 mai 2022 , Moyenne: 15 , Notes: [Java ,15].
Darmangua , Cristóbal , 19 août 1933 , 13 mai 2022 , Moyenne: 3 , Notes: [C# ,3].
Leri , Juan Miguel , 22 février 1935 , 13 mai 2022 , Moyenne: 1 , Notes: [C ,1].
Sont reçus: Juñes .
Sont recalés: Leri , Darmangua .
```

6. Modifier la classe `Étudiant` de sorte que l'on puisse consulter une note en particulier, par exemple de la manière suivante :

```
Etudiant jby = new Etudiant("Juan Bautista", "Juñes", 18, 3, 1934);
jby.add(new Note("Java", 15));
jby.add(new Note("C#", 1));
jby.add(new Note("C", 12));
Note csharp = jby.getNote("C#");
System.out.println(jby.getNom()+" a obtenu la note de "+csharp.getNote()+" en "+csharp.getIntitule());
```

qui pourrait produire quelque chose comme :

```
Juñes a obtenu la note de 1 en C#.
```

7. Modifier la classe `Cursus` de sorte que l'on puisse consulter toutes les notes d'un cours en particulier pour tous les étudiants, par exemple de la manière suivante :

```
Etudiant jby = new Etudiant("Juan Bautista", "Juñes", 18, 3, 1934);
Etudiant cd = new Etudiant("Cristóbal", "Darmangua", 19, 8, 1933);
Etudiant jml = new Etudiant("Juan Miguel", "Leri", 22, 2, 1935);
jby.add(new Note("Java", 15));
jby.add(new Note("C#", 1));
cd.add(new Note("C#", 3));
jml.add(new Note("C", 1));
Cursus c = new Cursus("PISE");
c.add(jby);
c.add(cd);
c.add(jml);
System.out.println("Les notes de C# pour le master "+c.getIntitule()+" sont: "+c.getNotes("C#"));
```

ce qui pourrait donner :

```
Les notes de C# pour la master PISE sont: juñes=1; darmangua=3.
```

8. Comment s'assurer qu'un étudiant ne puisse être présent deux fois dans le même cursus ?