

# NFA032 : TP n°11, héritage

15 mai 2018

## 1 UE et auditeurs

Un auditeur du CNAM peut être inscrit à plusieurs unités d'enseignement (UE). Chaque UE est validée si l'auditeur obtient une note supérieure à 10 à l'une des deux sessions d'examen.

On va représenter les auditeurs et leurs inscriptions aux UE au moyen de deux classes appelées `Auditeur` et `Inscription`.

Voici les caractéristique de la classe `Auditeur` :

- Données : nom et numéro INE (numéro d'auditeur), valeurs auxquels l'auditeur est ou a été inscrit.
- Un constructeur construisant un élève sans aucune inscription.
- Une méthode `ajouterInscription` permettant d'ajouter une inscription à l'élève.
- Une méthode `toString` renvoyant une chaîne avec toutes les données de l'objet y compris ses inscriptions.

Voici les caractéristique de la classe `Inscription` :

- Données : le code de l'unité d'enseignement, l'année et le semestre d'inscription, les notes des deux sessions.
- Un constructeur permettant de créer l'objet au moment de l'inscription (les notes ne sont pas connues).
- Une `ajouterNote` qui permet de renseigner les notes de première et deuxième session.
- Une méthode `estValidee` renvoyant un booléen et disant si l'UE est acquise par l'auditeur.
- Une méthode `toString`

1. Écrivez les deux classes `Auditeur` et `Inscription`.
2. Écrivez une méthode `main` avec le cas d'un auditeur nommé Martin, ayant obtenu 12 en première session de NFA031 au premier semestre 2016/2017 et inscrit sans encore de note à NFA032 au second semestre 2017/2018.

## 2 Contrôle continu

Certaines valeurs prennent en compte une note de contrôle continu dans l'évaluation. Par exemple, NFA032 comporte une note de contrôle continu qui compte pour 30% dans la note finale. D'autres valeurs utilisent un autre coefficient pour le contrôle continu (par exemple 25%).

1. Ecrivez une sous-classe `InscriptionCC` de `Inscription` qui représente les inscriptions aux UE avec contrôle continu. Elle doit contenir une méthode qui permette de renseigner la note de contrôle continu. Il faut également redéfinir la méthode `estValidee` pour tenir compte du contrôle continu.

2. Peut-on mettre un objet instance de `InscriptionCC` dans une variable de type `Inscription` ? Dans une case de tableau de type `InscriptionCC` ?
3. On propose le code suivant :

---

```
public static void main(String[] args) {
    Inscription[] tab = new Inscription[3];
    double note;
    Scanner scan = new Scanner(System.in);
    tab[0] = new Inscription("NFA003",2017,1);
    tab[1] = new InscriptionCC("NFA031",2017,1,0.3);
    tab[2] = new Inscription("NFA008",2017,2);
    for (int i=0; i<3; i++){
        System.out.println("Entrez la note de " + tab[i].getCode());
        note = scan.nextDouble();
        tab[i].ajouterNote(note, 1);
    }
    for (int i=0; i<3; i++){
        if (tab[i].estValidee())
            System.out.println("Validée: " + tab[i]);
    }
    scan.close();
}
```

---

Ce code est correct, il compile et s'exécute. Essayez de deviner ce qui va se passer si l'on tape 10 comme note pour chacune des trois valeurs. Puis exécutez le programme pour vérifier s'il s'exécute comme vous le pensez.

4. Exécutez le code pour vérifier si vous avez bien deviné.
5. Expliquez pourquoi le NFA031 n'est pas validée avec une note de 10.

### 3 Contrôle continu (suite)

Etudiez le code suivant dans lequel la méthode `main` a été modifiée, où l'on cherche maladroitement à enregistrer une note de contrôle continu pour NFA031.

---

```
public static void main(String[] args) {
    Inscription[] tab = new Inscription[3];
    double note;
    Scanner scan = new Scanner(System.in);
    tab[0] = new Inscription("NFA003",2017,1);
    tab[1] = new InscriptionCC("NFA031",2017,1,0.3);
    tab[2] = new Inscription("NFA008",2017,2);
    for (int i=0; i<3; i++){
        System.out.println("Entrez la note de " + tab[i].getCode());
        note = scan.nextDouble();
        tab[i].ajouterNote(note, 1);
        if (i==1){
```

```

        System.out.println("Entrez_la_note_de_contrôle_continu_de_"
            + tab[i].getCode());
        note = scan.nextDouble();
        tab[i].ajouterNoteCC(note);
    }
    if (tab[i].estValidee())
        System.out.println("Validée:_" + tab[i]);
    }
    scan.close();
}

```

---

1. Expliquez pourquoi ce code ne va pas être accepté par le compilateur. Quelle erreur de compilation sera affichée ?
2. Vérifiez que vous avez bien répondu en demandant la compilation de ce code.

## 4 Contrôle continu (fin)

On voudrait, dans la boucle de main, enregistrer selon le cas soit la note d'examen seule, soit la note d'examen et celle de contrôle continu. On pourrait être tenté de résoudre le problème de la façon suivante :

- Enregistrer la note d'examen.
- Tester si l'objet dans `tab[i]` est instance de `InscriptionCC`.
- Si oui, caster l'objet vers le type `InscriptionCC` pour appeler la méthode `ajouterNoteCC`, ce qui s'écrit comme suit :
 

```
((InscriptionCC) tab[i]).ajouterNoteCC(note);
```
- Si `tab[i]` n'est pas instance de `InscriptionCC`, ne rien faire.

Il est possible d'écrire cet algorithme et il fonctionne, mais ce n'est pas comme cela qu'on écrit un programme en style orienté objet : on ne va pas tester dans la classe `Auditeur` le type d'instance des objets pour faire des opérations différentes selon le cas, mais on va appeler une méthode de l'objet qui fera le travail. Cet appel aura des effets différents parce que la méthode appelée sera redéfinie dans la sous-classe.

Concrètement, ici, il faut :

1. ajouter à `Inscription` une méthode `saisirResultat` qui va lire au clavier la note d'examen et l'enregistrer dans l'objet
2. redéfinir la méthode `saisirResultat` dans la classe `InscriptionCC` pour qu'elle lise au clavier et enregistre dans l'objet les deux notes (examen et contrôle continu).
3. appeler la méthode `saisirResultat` dans le main de la classe `Auditeur`

Faites ces trois modifications et testez le programme obtenu.

## 5 À rendre

Vous devez rendre l'archive zip du projet eclipse réalisé. Si vous avez le temps, faites les questions optionnelles qui suivent.

## **6 Question optionnelle 1**

Modifiez la méthode `toString` de la classe `Auditeur` pour qu'elle indique les UE obtenues.

## **7 Question optionnelle 2**

Ajoutez au projet une classe des unités de TP qui n'ont pas de note d'examen, seulement une note de contrôle continu.