

<b>Module</b> : Conception orientée objet et programmation Java		<b>Documents autorisés</b> : Non
<b>Enseignants</b> : Equipe Java		<b>Nombre de pages</b> : 4
<b>Date</b> : 08/01/2015	<b>Heure</b> : 11:15	<b>Durée</b> : 1h30
<b>Classes</b> : 3INFO A,3 SIGMA 4INFINI		<b>Session</b> : Principale

## GESTION D'UNE UNIVERSITE

Nous souhaitons développer une application simplifiée de gestion d'une université. Pour réaliser notre application, nous vous demandons de compléter les classes présentées ci-dessous.

**Les parties à compléter sont numérotées de 1 à 18.**

**You devez mettre dans vos copies le numéro correspondant à chaque code ajouté //TODO XX :**

**You devez mettre TOUTE la méthode à compléter y compris la signature.**

### Travail demandé :

En précisant le numéro de chaque partie à ajouter :

- 1) Complétez la classe **Etudiant** avec les méthodes **equals**, **hashCode** et **toString**, deux étudiants sont égaux en cas d'égalité de leur *cin* et *nom*. // TODO 1 //
- 2) Complétez la classe **EnsembleEtudiant** regroupant un ensemble d'Etudiants dans un **HashSet**. Complétez les méthodes de cette classe sachant qu'elle implémente l'interface **IEtudiantService**. // TODO 2 -> TODO 10 //
- 3) Complétez la classe **Université** qui regroupe pour chaque Classe l'ensemble de ses Etudiants dans un **HashMap**. // TODO 11 -> TODO 18 //

Remarque 1 : On suppose que la classe **Classe** contient déjà les méthodes **equals**, **hashCode** et **toString**. Deux Classes sont égales en cas d'égalité de leurs libellés.

**Remarque 2 :Toutes les classes sont présentées dans l'ANNEXE ci-dessous.**

### ANNEXE :

```
public class Etudiant {
    private String nom;
    private int cin;
    private int age;
    private double moyenne;
    public Etudiant() {
    }
    public Etudiant(String nom, int cin,
int age, double moyenne) {
        this.nom = nom;
        this.cin = cin;
        this.age = age;
        this.moyenne = moyenne;
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom) {
        this.nom = nom;
    }
    public int getCin() {
```

```
        return cin;
    }
    public void setCin(int cin) {
        this.cin = cin;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public double getMoyenne() {
        return moyenne;
    }
    public void setMoyenne(double moyenne) {
        this.moyenne = moyenne;
    }
}
// TODO 1 - à compléter //
/* equals, hashCode et toString */
(1 point) (0,5 point) (0,5 point)
```

```
public interface IEtudiantService<T> {
    public void ajouterEtudiant(T t);
    public void supprimerEtudiant(T t);
    public boolean chercherEtudiant(Etudiant e);
    public boolean chercherEtudiant(int cin);
    public void afficher();
    public TreeSet<Etudiant> trierEtudiantsNom();
}
```

```
public class EnsembleEtudiants // TODO 2 - à compléter //(0,5 point)
{

    public Set<Etudiant> etudiants;

    public EnsembleEtudiants() {
        // TODO 3 - à compléter //(0,5 point)
    }

    @Override
    public void ajouterEtudiant(Etudiant e) {
        // TODO 4 - à compléter //(0,5 point)
    }

    @Override
    public void supprimerEtudiant(Etudiant e) {
        // TODO 5 - à compléter // (0,5 point)
    }

    @Override
    public boolean chercherEtudiant(Etudiant e) {
        // TODO 6 - à compléter // (0,5 point)
    }

    @Override
    public boolean chercherEtudiant(int cin) {
        // TODO 7 - à compléter // (1 point)
        /* avec l'API stream */
    }

    @Override
    public void afficher() {
        // TODO 8 - à compléter //
        /* avec l'API stream */ (1 point)
    }

    @Override
    public TreeSet<Etudiant> trierEtudiantsNom() {
        // TODO 9 - à compléter Trier par nom//
        /* Utiliser Lambda Expression ou API Stream ou Java 7 */ (1,5 point)
    }

    public double sommeDesMoyennes() {
        // TODO 10 - à compléter//
        /* avec l'API stream */(2 points)
    }
}
```

```

public class Classe {

    private String libelle;
    private int capacite;

    public Classe(String libelle, int capacite) {
        this.libelle = libelle;
        this.capacite = capacite;
    }

    public String getLibelle() {
        return libelle;
    }

    public void setLibelle(String libelle) {
        this.libelle = libelle;
    }

    public int getCapacite() {
        return capacite;
    }

    public void setCapacite(int capacite) {
        this.capacite = capacite;
    }
}

```

```

public class University {

    private String nom;
    private Map<Classe, EnsembleEtudiants> map;

    public University() {
        // TODO 11 - à compléter //(0,5 point)
    }

    public void ajouterClasse(Classe c) {
        // TODO 12 - à compléter //(0,5 point)
    }

    public void ajouterEtudiant(Classe c, Etudiant e) {
        // TODO 13 - à compléter //(1,5 point)
    }

    public void deplacerEtudiant(Etudiant e, Classe classeDestination) {
        // TODO 14 - à compléter //(1,5 point)
        /* l'étudiant doit être supprimé de sa première classe */
    }

    public void permuterEtudiant(Etudiant e1, Etudiant e2) {
        // TODO 15 - à compléter //(1 point)
        /* les deux étudiants doivent existés */
    }

    public void afficherUniversity() {
        // TODO 16 - à compléter //(1,5 point)
        /* Utiliser Lambda Expression ou API Stream ou Java 7*/
    }
}

```

```
public void afficherNomEtudiantsAge(Classe c) {  
    // TODO 17 - à compléter//(1,5 point)  
    /* avec l'API stream */  
    /* Afficher les noms des étudiants d'une classe dont  
       leur âge>20 et      <23 */  
}
```

```
public List<Etudiant> meilleursEtudiants() {  
    // TODO 18 - à compléter//(2 points)  
    /* avec l'API stream */  
    /* Retourner les dix premiers étudiants selon  
       leurs moyennes dans une liste */  
}
```