

Pour cette séance vous ferez le dépôt de votre travail sous la forme **d'un seul fichier**, une archive **tar**, propre, ne contenant que vos sources `.java` (et aucun `.class`). Chacun fera individuellement son dépôt (pas de binômes cette fois ci). On rappelle la commande de création d'une archive à partir d'une liste de fichiers : `tar cvf archive.tar fichier1 fichier2 etc.` Et de façon similaire, à partir d'un dossier : `tar cvf nomArchive.tar dossierContenantLesSources/`

Exercice 1

1. Reprenez les codes des classes `Test` et `Etudiant` suivantes que nous allons modifier par la suite :

```
public class Test{
2   public static void main(String[] args){
      /* A COMPLETER */
4  }
}

1 public class Etudiant{
    final String prenom; // le prenom,
3    final String nom;   // le nom et
    double note;         // la note de l'etudiant (sur 20).

    static int nombreDEtudiants = 0; // le nombre d'etudiants dans la promo
7    static double sommeDesNotes = 0; // la somme des notes des etudiants

9    /* A COMPLETER */
}
```

2. Rappelez-vous ce que signifient les mots clefs `static` et `final`. Que pouvez vous dire de la visibilité des attributs ? (répondre dans un commentaire dans le fichier `Test.java`).

3. Ajoutez à la classe `Etudiant` un constructeur `Etudiant(String prenom, String nom, double note)` qui :

- initialise chacun des trois attributs (`prenom`, `nom`, `note`) avec l'argument approprié,
- augmente `nombreDEtudiants` de 1 et ajoute à `sommeDesNotes` la note de l'étudiant courant.

4. Testez ce constructeur en ajoutant les lignes suivantes dans le `main` de la classe `Test`.

```
Etudiant e1 = new Etudiant("Luke", "Skywalker", 8.25);
2 Etudiant e2 = new Etudiant("Leia", "Organa", 11.75);
System.out.println("nb d'etudiants: "+Etudiant.nombreDEtudiants);
4    // nb d'etudiants: 2
System.out.println("somme des notes: "+Etudiant.sommeDesNotes);
6    // somme des notes: 20
```

Posez vous la questions suivante : peut-on accepter que dans ce test on introduise une instruction `Etudiant.nbEtudiant = -10`; Comment peut-on concilier avec la volonté de vouloir autoriser d'afficher cette valeur ? Prenez les mesures qui correspondent à votre réflexion.

5. Créez un troisième `Etudiant` avec votre prénom, votre nom et la note que vous estimez pour votre travail à ce TP ; puis affichez à nouveau les variables statiques `nombreDEtudiants` et `sommeDesNotes`.

6. Ajoutez à la classe `Etudiant` une méthode `void afficher()` qui affiche les informations relative à l'étudiant courant sous le format "`<nom> <prenom> : <note>`".

7. Testez en ajoutant les lignes suivantes dans `main`.

```

e1.afficher(); // Skywalker Luke: 8.25
2 e2.afficher(); // Organa Leia: 11.75

```

A titre d'exercice de style, écrivez une seconde méthode `afficher2` qui serait elle déclarée statique et ferait la même chose qu'`afficher`. (Déterminez sa signature exacte, et tester la)

8. Écrivez une méthode `boolean estAdmis()` qui renvoie `true` si l'étudiant est admis, c'est-à-dire si sa note est supérieure ou égale à 10. (Testez là).

9. Écrivez une méthode `static double moyenne()` qui renvoie la moyenne de tous les étudiants. (Testez là).

10. Écrivez une méthode `boolean meilleurQueLaMoyenne()` qui renvoie `true` si la note de l'étudiant est supérieure à la moyenne de la promotion. (Testez là).

11. Ajoutez une méthode `void modifierNote(double nouvelleNote)` qui remplace l'ancienne `note` de l'étudiant par son argument. (Attention à ne pas oublier de modifier `sommeDesNotes` de façon adéquate) Testez, par exemple avec les lignes suivantes :

```

e2.modifierNote(19.5);
2 e2.afficher();
    // Organa Leia: 19.5
4 System.out.println("moyenne: "+Etudiant.moyenne());
    // moyenne: 14
6    // [Si la promotion ne contient que Luke, Leia et vous avec 14.25]

```

12. Relisez votre travail en étant clair sur ce qui est considéré `public` et `private` si vous n'y avez pas totalement prêté attention. Déposez déjà ce travail en respectant strictement les consignes.

Exercice 2

1. Créez une nouvelle classe `Trio` comme suit.

```

public class Trio {
2     private Etudiant[] membres;
    /* A COMPLETER */
4 }

```

2. Ajoutez un constructeur `Trio(Etudiant e1, Etudiant e2, Etudiant e3)` qui initialise l'attribut `membres` à un tableau de trois éléments contenant les trois étudiants donnés en argument.

3. Dans `Test` créez un trio contenant Luke, Leia et vous-même.

4. Dans la classe `Trio` écrivez une méthode `Etudiant premier()` qui renvoie celui des trois qui a la meilleure note. (Testez la)

5. Ajoutez une méthode `int classement(String prenom, String nom)` qui renvoie le classement à l'intérieur du trio de l'étudiant dont le nom et le prénom sont donnés en argument : s'il a la meilleure note du trio, renvoyer 1, s'il a la deuxième meilleure note, renvoyer 2, etc. Si aucun étudiant du trio ne porte ces nom et prénom, renvoyer 0. (Précisez en commentaire le choix que vous faites pour les ex-aequo)

6. Ajoutez une méthode `double moyenne()` qui renvoie la moyenne des notes des membres du trio et une méthode `boolean meilleurQueLaMoyenne()` qui renvoie `true` si la moyenne du trio est supérieure à la moyenne de la promotion.