Exercice 1

1. Créer une classe `Test` contenant le code suivant.

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

2. Créer une autre classe `Etudiant` contenant le code suivant.

```
class Etudiant
{
  final String prenom; // le prenom,
  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 classe.
  static double sommeDesNotes = 0; // la somme des notes des etudiants.

  /* A COMPLETER */
}
```

3. Que signifie les mots clefs `static` et `final` (Répondre dans un commentaire dans le fichier Test.java).

4. Ajouter à la classe `Etudiant` un constructeur `Etudiant(String prenom, String nom, double note)` qui
   - initialise chacun des trois attributs (`nom`, `prenom`, `note`) avec l'argument approprié
   - augmente `nombredEtudiants` de 1 et ajoute à `sommeDesNotes` la note de l'élève courant.

Tester le constructeur en ajoutant les lignes suivantes dans le `main` de la classe `Test`.

```
Etudiant victor = new Etudiant ("Victor", "Marsault", 8.25);
Etudiant victoria = new Etudiant ("Victoria", "Marceau", 11.75);
System.out.println("nb d'etudiants: "+Etudiant.nombredEtudiants);
    // nb d'etudiants: 2
System.out.println("somme des notes: "+Etudiant.sommeDesNotes);
    // somme des notes: 20
```

Créer un troisième `Etudiant` avec votre nom, votre prénom et la note 14.25 ; puis afficher à nouveau les variables statiques `nombredEtudiants` et `sommeDesNotes`.

5. Ajouter à la classe `Etudiant` une méthode `void affiche()` qui affiche les informations relative à l'élève courant (`this`)) sous le format "<nom> <prenom> : <note>".

Tester en ajoutant les lignes suivantes dans `main`.

```
    victor.affiche(); // Marsault Victor: 8.25
    victoria.affiche(); // Marceau Victoria: 11.75
```

6. Ecrire une méthode `boolean passage()` qui renvoie `true` si l'élève peut passer dans la classe supérieure, c'est-à-dire si sa note est supérieure ou égale à 10.

Tester.

7. Ecrire une méthode `static double moyenne()` qui renvoie la moyenne de tous les élèves.

Tester.

8. Ecrire une méthode `boolean meilleurQueLaMoyenne()` qui renvoie `true` si la note de l'élève est supérieure à la moyenne de la classe.

Tester.

9. Ajouter une méthode `void modifieNote(double nouvelleNote)` qui remplace l'ancienne `note` de l'élève par `nouvelleNote`. (Attention à ne pas oublier de modifier `sommeDesNotes` de façon adéquate.)

Tester, par exemple avec les lignes suivantes.

```
  victoria.modifieNote(19.5);
2 victoria.affiche();
      // Marceau Victoria: 19.5
4 System.out.println("moyenne:␣"+Etudiant.moyenne());
      // moyenne: 14
6     // [Si la classe ne contient que Victor, Victoria et vous-meme.]
```

Exercice 2 (Facultatif)

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

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

2. Ajouter un constructeur `Trio(Etudiant e1, Etudiant e2, Etudiant e3)` qui initialise l'attribut `membres` à un tableau de trois éléments contenant les trois `Etudiant` donnés en argument.
Tester dans `Test` en créant un trio contenant victor, victoria et vous-même.

3. Ecrire une méthode `Etudiant premier()` qui renvoie l'élève du trio ayant la meilleure note.

Tester sur le trio créé à la question précédente ; la fonction doit renvoyer victoria.

4. Ajouter une méthode `int classement(String prenom, String nom)` qui renvoie le classement à l'intérieur du trio de l'élève dont les nom et prénom sont donnés en argument. S'il a la meilleure note du trio, renvoyer 1, s'il est deuxième renvoyer 2, etc. Si aucun élève du trio ne porte ces nom et prénom, renvoyer 0.

Tester.

5. Ajouter 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 classe.

Tester.