

Corrigés - Série n°3 de TP

Exercice 5

```
class Etudiant
{ final static int coef1= 4 ;
  final static int coef2= 3 ;      // constantes de classe
  final static int coef3= 4 ;
  float note1, note2, note3, moy;
  long numero; String Nom, Prenom;

  void Lire ()
  { Scanner e = new Scanner (System.in);
    System.out.println (" Introduire les infos d'un étudiant");
    numero = e.nextLong(); Nom = e.next(); Prenom = e.next();}

  void Afficher ()
  { System.out.print (numero + "    " + Nom + "    " + Prenom + "    " +
    moy ) ;}

  void AffecterNotes (float n1, float n2, float n3)
  { note1 = n1; note2 = n2; note3 = n3; }

  void CalculerMoy ()
  { moy = (note1*coef1 + note2*coef2 + note3*coef3)/ (coef1 + coef2+ coef3);} } //
fin de la classe Etudiant
```

// définition de l'énumération Mention

```
enum Mention
{ Excellent (18), Très Bien (16), Bien (14), Assez Bien (12), Passable (11) ;
  float moymin ; // on peut déclarer de type int, moymin correspond à la valeur
                  spécifiée entre parenthèses
  Mention (float m) {moymin = m ;} // constructeur qui affecte automatiquement
                                  les valeurs de moymin
```

```
import java.util.Scanner;
class ProgEtudiant
```

```
{ Scanner e = new Scanner (System.in); float n1, n2, n3;
  System.out.println (" donnez le nombre d'étudiants"); int n = e.nextInt();
  Etudiant V[ ] = new Etudiant [n]; // création du tableau de références
  for (i= 0; i<n; i++)
  { // Création de l'objet Etudiant référencée par V[i]
    V[i] = new Etudiant ();
    V[i].Lire(); // introduction des données
    System.out.println (" Introduire les notes de l'étudiant");
    n1= e.nextFloat(); n2= e.nextFloat(); n3 = e.nextFloat();
    V[i]. AffecterNotes (n1, n2, n3);
```

```

V[i].CalculerMoy();          V[i].Afficher();
/* pour afficher la mention de l'étudiant, il faut comparer sa moyenne
avec moymin dans la classe enum Mention */.
for (Mention m : Mention.values()) // m parcourt toutes les valeurs de Mention
                                     dans l'ordre
    if (V[i].moy >= m.moymin) {System.out.println (m); break ;} // on affiche la mention }

```

Question supplémentaire

```

// tri du vecteur par ordre décroissant des valeurs de moyennes (tri par bulles)
boolean permut = true ; Etudiant E;
while (permut == true)
    { permut = false;
      for (i= 0; i<n-1; i++)
          { if (V[i].moy < V[i+1].moy )

              { E = V[i]; V[i] = V[i+1]; V[i+1] = E; permut = true;}}

// Reprendre l'affichage, on peut utiliser une autre forme de la boucle for
for (Etudiant e : V) // e de type Etudiant prend les valeurs de V[0], V[1], ... V[n-1]
    {e. Afficher();
     /* pour afficher la mention de l'étudiant, il faut comparer sa moyenne avec moymin
dans la classe enum Mention */.

    for (Mention m : Mention.values()) // m parcourt toutes les valeurs de Mention dans
l'ordre
        if (e.moy >= m.moymin) System.out.println (m); // on affiche la mention
    }}}

```

Exercice 6

```

class Joueur { int numero, String nom, int score ;
// constructeur
Joueur (int num, String n) { numero = num; nom = n; score = 0;}
void Incrementer () { score ++; }
void afficher ()
{ System.out.println ("le joueur numéro" + numero + "de nom" + nom + "a obtenu un score de" +
score) ; }
}

2/ import java.util.Scanner ;
import java.util.Random ;
Class ProgramJeu
{ public static void main(String arg[])

    { Joueur J1, J2 ; int num; String nom; int score ; int i, val, x;
      Scanner e = new Scanner (System.in) ;
      Random r = new Random() ;
      System.out.println ("donnez la description du 1er joueur") ;
      num= e.nextInt() ; nom = e.next() ; J1 = new Joueur (num, nom) ;
      System.out.println ("donnez la description du 2eme joueur") ;
      num= e.nextInt() ; nom = e.next() ; J2 = new Joueur (num, nom) ;
      System.out.println ("le joueur 1 doit introduire une valeur") ;
      val = e.nextInt();

```

```

        for (i= 1 ; i<= 10 ; i++)
            { x = r.nextInt() ;
              if (x >= val-1000 && x<= val+1000) J1.Incrementer(); }

        System.out.println ("le joueur 2 doit introduire une valeur") ;
        val = e.nextInt();
        for (i= 1 ; i<= 10 ; i++)
            { x = r.nextInt() ;
              if (x >= val-1000 && x<= val+1000) J2.Incrementer(); }

        if (j1.score > j2.score) { System.out.println ("le joueur 1 a gagné") ; J1.Afficher() ;}
        else if (j2.score > j1.score) { System.out.println ("le joueur 2 a gagné") ; J2.Afficher() ;}
        else System.out.println ("scores égaux") ;

3/ // tableau de joueurs, partie avec 20 joueurs
   Joueur V[] = new Joueur [20] ;
   for (i= 0 ; i< 20 ; i++)
       { System.out.println ("donnez la description du joueur" + (i+1)) ;
         num= e.nextInt() ; nom = e.next() ; V[i]= new Joueur (num, nom) ;
         System.out.println ("le joueur numéro" + (i+1)+ " doit introduire une valeur") ;
         val = e.nextInt();
         for (j= 1 ; j<= 10 ; j++)
             { x = r.nextInt() ;
               if (x >= val-1000 && x<= val+1000) V[i].Incrementer(); }
       }

4/ /* tri du vecteur, on peut utiliser l'une des méthodes de tri usuelles, il suffit de permuter les
références */
int imax ; Joueur X ;
for (i= 0 ; i< 19 ; i++)
    { imax = i;
      for (j = i+1 ; j<20; j++)
          { if (V[j].score > V[imax].score) imax = j;}
      if (imax != i) // permuter les références
          { X= V[j] ; V[j] = V[imax] ; V[imax] = X ; }
    }

// affichage dans l'ordre décroissant
System.out.println ("Voici les scores des joueurs dans l'ordre décroissant") ;
for (i= 0 ; i< 20 ; i++) V[i].Afficher() ; } }

5/ si l'attribut score était declare « private », on doit définir une méthode (accesseur getScore) qui
retourne sa valeur dans la classe Joueur

class Joueur
{ ...
  private int score ;
  ...
  public getScore() {return (score) ; } ...}
dans le programme de tri, on remplace la condition if (V[j].score > V[imax].score) par
if (V[j].getScore() > V[imax].getScore())

```

Question: Afin d'améliorer la lisibilité de la méthode main et la réutilisation du code, Reprendre l'exercice en ajoutant une méthode **PartieDejeu** dans la classe Joueur. La méthode implémente une partie de jeu pour un joueur et incrémente son score s'il ya lieu.
