Université Abou Bakr Belkaïd – Tlemcen - Faculté des Sciences - Département d'Informatique

2018- 2019 **2ème année licence informatique**

EF – Programmation Orientée Objet

Documents interdits **Durée : 1h30**

Exercice 1 (04 points)

Soit la classe A

```
class A {
        int a
        A(int a){ this.a=a ;}
 }
```

1. Modifier la classe A en la transformant en Singleton. C'est-à-dire A ne doit instancier qu'un seul objet.
2. Dans une classe principale, écrire l'instruction qui permettra de retourner cet unique objet de A.

Exercice 2 (06 points)

Soient les deux classes suivantes :

```
public class Nombre {
        private int valeur;
        public Nombre(int d) {  valeur = d;  }
        public double racineCarree() {  return Math.sqrt(valeur);  }
}
```

```
public class TestRacineCarreeException {
        public static void main(String[] args) {
                java.util.Scanner clavier = new java.util.Scanner(System.in);
                int entier = clavier.nextInt();
                Nombre d = new Nombre(entier);
                System.out.println(d.racineCarree());          }
}
```

Complétez le programme ci-dessus pour que les deux (02) erreurs susceptibles de se produire soient gérées.

1. Si la valeur de "**entier**" (qui est saisi) n'est pas de type "**int**", alors le programme interceptera l'exception "**InputMismatchException**" et affichera le texte : « Il faut saisir un entier ! »
2. Si **entier** est **négatif**, alors le programme interceptera l'exception **"NegativeException"** que vous devez d'abord créer et lever. Il affichera ensuite, le texte retourné par la méthode "**toString()"** de "**NegativeException".**
   Par exemple, si entier = -9 alors on aura : « NegativeException: -9 est une valeur négative»

Exercice 3 (10 points)

Soit l'interface Comparable contenant la méthode comparer (Object o) qui permet de comparer des séries selon leurs nombres de saisons et les des films selon leurs durées.

```
public interface Comparable {
     int comparerA(Object o);}
```

Complétez les classes suivantes :

```
_____Video implements Comparable {
   // Constantes de classes à utiliser dans la comparaison de vidéos
    _____ int PLUSPETITE=1;
    _____ int PLUSGRANDE=-1;
    _____ int EGALE=0;

    private String titre;
    //Consturcteur
    _____
    // Getter et setter
    _____
    _____
    _____toString(){
        return "Informations sur "+_____+  // Film ou serie
                ": \n" +"\tTitre: "+_____+"\n" ;
    }
}
```

```
public class Film extends Video {
     private double durée;
   // Contructeur
    _____________________________________
    // Getter et Setter
    _____________________________________
    _____________________________________
   // Une autre Méthode
    _____________________________________
    _____________toString(){
    _____________________________________
    }
}
```

```
public class Serie extends Video {
      private int nombreSaison;
   // Contructeur
    _____________________________________
    // Getter et Setter
    _____________________________________
    _____________________________________
   // Une autre Méthode
    _____________________________________
    _____________toString(){
    _____________________________________
    }
}
```

```
_____________________
public class Vidéothèque {
    ArrayList<Video> videos = new ArrayList<Video>();
    public void ajouter (Video v){
        _________________
    }

    public void listerFilm(){
        _____________________________________
    }
}
```

```
public class PP {
    public static void main(String[] args) {
        Video v1 = new Film ("Le seigneur des anneaux",2.58);
        Video v2 = new Film ("Terminator",1.48);
        Video v3 = new Serie ("Dr House",8);
        Vidéothèque V=new Vidéothèque();
        V.ajouter(v1);        V.ajouter(v2);        V.ajouter(v3);        V.listerFilm();
        // Comparer v1 et v2 en affichant le résultat
        _____________________________________
    }
}
```

Bon Courage !

Corrigé

Exercice 1 (04 points)

Soit la classe A

```
class A {
        private static A instance;                          1
        int a
        private A(int a){ this.a=a ;}                       0.5
        public static A getInstance(int a) {                1
                if (instance = null)
                        instance = new A(a);
                return instance;                            0.5
        }
 }
```

```
class PP {
        public static void main (String [] args) {
                A a=A.getInstance (5);                      1
```

Exercice 2 (06 points)

```
public class Nombre {
  private int valeur;

  public Nombre(int d) {
    this.valeur = d;
  }

  public double racineCarree() throws NegativeException {   0.5
    if (this.valeur<0) throw new NegativeException(valeur); 1

    return Math.sqrt(this.valeur);
    }
}
```

```
public class NegativeException extends Exception{           0.5
   private int n;                                           0.5

  public NegativeException( int n) { this.n= n;}            0.5
  public String toString() {
        return super.toString()+" : "+n+ " est un nombre négatif";}   1
}
```

```
public class TestRacineCarreeException {
  public static void main(String[] args) throws NegativeException {
    java.util.Scanner clavier = new java.util.Scanner(System.in);

    try {                                                   0.5
       int entier = clavier.nextInt();
       Nombre d = new Nombre(entier);
       System.out.println(d.racineCarree());
    }
    catch(NegativeException e){System.out.println(e);}      0.5
    catch (java.util.InputMismatchException e) {            0.5
        System.out.println("Il faut saisir un entier");};   0.5
  }
}
```

Exercice 3 (10 points)

Classe Video (2.75 points)

```
public abstract class Video implements Comparable {                    0.25
   // Constantes de classes à utiliser dans la comparaison de vidéos
    public static final int PLUSPETITE=1;                              0.25
    public static final int PLUSGRANDE=-1;                             0.25
    public static final  int EGALE=0;                                  0.25

    private String titre;
    //Consturcteur
    public Video (String titre){                                       0.5
          this.titre=titre;
    }
    // Getter et setter
    public String getTitre() {                                         0.25
          return titre;
    }

    public void setTitre(String titre) {                               0.25
       this.titre = titre;
    }

     public String toString(){                                         0.25
        return "Informations sur "+ getClass().getName() +  // Film ou serie  0.25
                ": \n" +"\tTitre: "+ titre +"\n" ;                     0.25
    }
}
```

Classe Film (2.75 points)

```
public class Film extends Video {
     private double durée;
   // Contructeur
    public Film(String titre, double durée){                           0.5
       super(titre);
       this.durée=durée;
    }
    // Getter et Setter
    public double getDurée() {                                         0.25
       return durée;
    }

    public void setDurée(double Durée) {                               0.25
       this.durée = durée;
    }
    // Une autre Méthode
    public int comparerA(Object a) {                                   0.25
       int etat=PLUSPETITE;
       Film ref=(Film)a;                                               0.25
       if (durée>ref.getDurée()){                                      0.5
          etat=PLUSGRANDE;
       }else if(durée==ref.getDurée()){
          etat=EGALE;
       }
         return etat;
    }
   public String toString(){                                           0.25
    return super.toString()+                                           0.25
             "\tDurée: "+durée+"\n" ;                                  0.25
    }
}
```

Classe Serie (2.75 points)

```
public class Serie extends Video {
      private int nombreSaison;
   // Contructeur
      public Serie(String titre, int nombreSaisons){                    0.5
            super(titre);
            this.nombreSaisons=nombreSaisons;
      }
    // Getter et Setter
      public int getNombreSaisons() {                                   0.25
            return nombreSaisons;
      }

      public void setnombreSaisons(int nombreSaisons) {                 0.25
            this.nombreSaisons = nombreSaisons;
      }
   // Une autre Méthode
     public int comparerA(Object a) {                                   0.25
            int etat=PLUSPETITE;
            Serie ref=(Serie)a;                                         0.25
            if (nombreSaisons>ref.getNombreSaisons()){                  0.5
                  etat=PLUSGRANDE;
            }else if(nombreSaisons==ref.getNombreSaisons()){
                  etat=EGALE;
                }
           return etat;
     }
     public String toString(){                                          0.25
            return super.toString()+                                    0.25
                "\tNombre de saisons: "+nombreSaisons+"\n" ;             0.25
     }
}
```

Classe Vidéothèque (1.25 point)

```
import java.util.*;                                                     0.25
public class Vidéothèque {
    ArrayList<Video> videos = new ArrayList<Video>();
    public void ajouter (Video v){
       videos.add(v);                                                   0.25
    }

    public void listerFilm(){
       for (Video v:videos)                                             0.75
          if (v instanceof Film) System.out.println(v);
    }
}
```

Classe PP (0.5 point)

```
public class PP {
    public static void main(String[] args) {
       Video v1 = new Film ("Le seigneur des anneaux",2.58);
       Video v2 = new Film ("Terminator",1.48);
       Video v3 = new Serie ("Dr House",8);
       Vidéothèque V=new Vidéothèque();
       V.ajouter(v1);        V.ajouter(v2);        V.ajouter(v3);        V.listerFilm();
       // Comparer v1 et v2 en affichant le résultat
      String s=" est égal à ";                                          0.5
       if (v1.comparerA(v2)==-1) s=" est plus long que ";
       else if (v1.comparerA(v2)==1) s=" est plus court que ";
       System.out.println(v1.getTitre() + s + v2.getTitre());
    }
}
```