Atelier 1 : Les exceptions

RT2

Exercice 1

```
class Temps
{
  //Attributs
  private int heures, minutes, secondes;
  //Constructeur
  public Temps(int h, int m, int s)throws TempsException
  {
     if((h>=0) && (h<=23)) heures = h; //Vérifier les heures
       else throw new TempsException();
    if((m>=0) && (m<=60)) minutes = m; //Vérifier les minutes
       else throw new TempsException();
    if((s>=0) && (s<=60)) secondes = s; //Vérifier les secondes
       else throw new TempsException();
  }
} 2
```

Exercice 1

```
public class TempsException extends Exception
{
  public String toString()
   {return(« temps invalide »);}
}

class Test
{public static void main(String [] args)
  { try
   {
     Temps t = new Temps(12, 12, 68);
   }
   catch(TempsException e)
   {
     System.out.println(e)
   }
}}
```

Exercice 2

```
class ExceptionPile extends Exception
{
  ExceptionPile(String s)
  {
    super(s);
  }
}
```

Exercice 2

```
Class Pile
{   private int[] P; //Attribut
int nbElem;  //Attribut
public Pile(int n) //constructeur
{
  P = new int[n];
}
void empile (int e) throws ExceptionPile
{
 if(!pilePleine()) {P[nbElem++] = e; }
 else throw new ExceptionPile(« Pile Pleine, on ne peut pas empiler »);
}

void depile()throws ExceptionPile
{
 if (!pileVide()) nbElem --;
 else throw new ExceptionPile(« Pile Vide, on ne peut pas dépiler »);
}
```

Exercice 2

```
void affiche() throws ExceptionPile
{
 if(pileVide())
  throw new ExceptionPile(« Pile Vide, rien à afficher »);
 else
  System.out.println(« contenu de la pile »);
for(int i=0; i<nbElem; i++)
System.out.println(P[i]);
}
boolean pilePleine()
{
 if (nbElem == tab.length)
  return(true);
 else   return(false); }
boolean pileVide()
{
 if (nbElem ==0) return(true);
 else   return(false); }
}//fin classe Pile
```

Exercice 2

```
public class Test
{ public static void main(String [] args)
   { Pile  P = new Pile(10);
   try //emplier 3 entiers
   {P.empile(1); P.empile(2); P.empile(3);}
   catch(ExceptionPile e) {System.out.println(e.getMessage());
   try //dépiler un élément
   {P.depile();}
   catch(ExceptionPile e) {System.out.println(e.getMessage());
   try //afficher la pile
   {P.affiche();}
   catch(ExceptionPile e) {System.out.println(e.getMessage());}
}
```

Exercice 3

```
public class Personne {
private String nom;
private int age;
public Personne (String n, int a) throws IllegalArgumentException {
   if (n == null)
   throw new IllegalArgumentException("pas de nom !");
   this.nom = n;
   if (a < 0)
   throw new IllegalArgumentException("nombre negatif de puces !");
   this.age = a;
   }
   public String toString() {
   return nom + " a " + a + " ans.";
}
}
```

Exercice 3

```
public class Test
{
public static void main(String[] args) {
   try {
     System.out.println("creation d'une première personne");
     Personne p1 = new Personne (« Ali", 4);
     System.out.println("le voici : " + p1);
     System.out.println("creation d'une deuxième personne");
     Personne p2 = new Toutou (« Salah", -11);
     System.out.println("le voici : " + p2);
   }
   catch (IllegalArgumentException e) {
     System.out.println("une personne ratée !! " + e);
   }
}
}
```