

## TD n°8 - Correction

### Exceptions

**Exercice 1** La méthode `parseInt` est spécifiée ainsi :

```
public static int parseInt(String s)
    throws NumberFormatException
```

Parses the string argument as a signed decimal integer. The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value. The resulting integer value is returned, exactly as if the argument and the radix 10 were given as arguments to the `parseInt(java.lang.String, int)` method.

Parameters:

s - a String containing the int representation to be parsed

Returns:

the integer value represented by the argument in decimal.

Throws:

NumberFormatException - if the string does not contain a parsable integer.

Utiliser cette fonction pour faire la somme de tous les entiers donnés en argument de la ligne de commande, les autres arguments étant ignorés.

**Correction :**

```
class Somme {
    public static void main(String[] args) {
        int somme = 0;
        for(int i=0;i<args.length;i++) {
            try { somme += Integer.parseInt(args[i]); }
            catch (NumberFormatException e) {}
        }
        System.out.println(somme);
    }
}
```

**Exercice 2** Écrire une classe `Pile` qui implémente une pile d'objets avec un tableau de taille fixe. On définira pour cela deux exceptions `PilePleine` et `pileVide`. On utilisera pour écrire les méthodes, l'exception `ArrayOutOfBoundsException` qui indique qu'on a tenté d'accéder à une case non définie d'un tableau.

Les champs de la classe seront :

```

private final static int taille = 10;
private Object [] pile;
private int pos;

```

Écrire une méthode main qui empile les arguments de la ligne de commande (du moins tant que c'est possible) et qui les réécrit dans l'ordre inverse.

On aura par exemple :

```

> java Pile 1 2 3 4 5 6 7 8 9 10 11 12 13
10 9 8 7 6 5 4 3 2 1

```

### Correction :

```

class PilePleine extends Exception{
    PilePleine(String s) { super(s); }
}

class PileVide extends Exception{
    PileVide(String s) { super(s); }
}

class Pile{
    private final static int taille = 10;
    private Object [] pile;
    private int pos;

    Pile() { pile=new Object[taille]; pos=0; }

    public void empile(Object o) throws PilePleine{
        try {
            pile[pos]=o;
            pos++;
        }
        catch(ArrayIndexOutOfBoundsException e){
            throw new PilePleine("Pile pleine!");
        }
    }

    public Object depile() throws PileVide{
        try {
            Object o = pile[pos-1];
            pos--;
            return o;
        }
        catch(ArrayIndexOutOfBoundsException e){
            throw new PileVide("Pile vide!");
        }
    }

    public static void main(String[] args){
        Pile p = new Pile();
        try {
            for(int i=0;i<args.length;i++) p.empile(args[i]);
        }
    }
}

```

```
    catch(PilePleine e) {};  
    try {  
        for(;;) System.out.print(p.depile()+" ");  
    }  
    catch(PileVide e) { System.out.println(); }  
}  
}
```

**Exercice 3** 1. Que fait le programme suivant ?

```
class Essai1Exception extends Exception{
    Essai1Exception (String s){
        super(s);
    }
}

class Essai2Exception extends Essai1Exception{
    Essai2Exception (String s){
        super(s);
    }
}

class Exn{
    static void throwEssais(int i) throws Exception {
        switch(i){
            case 1:
                System.out.println("Lancement de Essai1Exception");
                throw new Essai1Exception("Essai1Exception de throwEssais");
            case 2:
                System.out.println("Lancement de Essai2Exception");
                throw new Essai2Exception("Essai2Exception de throwEssais");
            default:
                System.out.println("Lancement de Exception");
                throw new Exception("Exception de throwEssais");
        }
    }

    public static void main(String[] args){
        for(int i=1; i<=3;i++){
            try{ throwEssais(i);}
            catch(Essai2Exception e){
                System.out.println("Catch Essai2 : "+e.getMessage());
            }
            catch(Essai1Exception e){
                System.out.println("Catch Essai1 : "+e.getMessage());
            }
            catch(Exception e){
                System.out.println("Catch Exception : "+e.getMessage());
            }
            finally{
                System.out.println("Finally de main");
            }
        }
    }
}
```

2. Et celui-ci?

```
class Essai1Exception extends Exception{
    Essai1Exception (String s){
        super(s);
    }
}

class Essai2Exception extends Essai1Exception{
    Essai2Exception (String s){
        super(s);
    }
}

class Exn{
    static void throwEssais(int i) throws Exception {
        switch(i){
            case 1:
                System.out.println("Lancement de Essai1Exception");
                throw new Essai1Exception("Essai1Exception de throwEssais");
            case 2:
                System.out.println("Lancement de Essai2Exception");
                throw new Essai2Exception("Essai2Exception de throwEssais");
            default:
                System.out.println("Lancement de Exception");
                throw new Exception("Exception de throwEssais");
        }
    }

    public static void main(String[] args){
        for(int i=1; i<=3;i++){
            try{ throwEssais(i);}
            catch(Essai1Exception e){
                System.out.println("Catch Essai1 : "+e.getMessage());
            }
            catch(Essai2Exception e){
                System.out.println("Catch Essai2 : "+e.getMessage());
            }
            catch(Exception e){
                System.out.println("Catch Exception : "+e.getMessage());
            }
            finally{
                System.out.println("Finally de main");
            }
        }
    }
}
```

**Correction :**

1. Lancement de Essai1Exception  
Catch Essai1 : Essai1Exception de throwEssais  
Finally de main

Lancement de Essai2Exception  
Catch Essai2 : Essai2Exception de throwEssais  
Finally de main

Lancement de Exception  
Catch Exception : Exception de throwEssais  
Finally de main

2. problème à la compilation :

```
Exn.java:22: exception Essai2Exception has already been caught
      catch(Essai2Exception e){
      ^
```

1 error