



## TP N° 6 (complément)

### Exercice 4 (Gestion des Exceptions)

Complétez le programme suivant pour que les erreurs susceptibles de se produire soient gérées jusqu'à ce que le calcul de factoriel soit effectivement mené à bout.

```
class FactorielleSansException {  
    public static void main(String[] argv) {  
  
        int i, n, factorielle = 1;  
        n = Integer.parseInt(argv[0]);  
  
        for (i = 2; i <= n; i++)    factorielle *= i;  
  
        System.out.println("Voilà la factorielle de " + n + " : " + factorielle);  
    }  
}
```

A partir de ce programme simple calculant la factorielle d'un entier donné sur la ligne de commande, essayez de gérer les cas d'erreurs :

1. Sur la ligne de commande, ne figure pas l'argument attendu.
2. L'argument n'est pas convertible en entier
3. L'argument est un entier négatif
4. L'argument est un entier trop grand

Vous devez modifier le programme pour que, dans chacun de ces cas, l'erreur soit attrapée par le programme et signalée à l'utilisateur.

1. S'il n'y a pas de paramètre sur la ligne de commande, il s'agit d'une exception de type **ArrayIndexOutOfBoundsException** ; on souhaite que le programme affiche par exemple : « Indiquez un nombre entier sur la ligne de commande », puis qu'il se termine.
2. Si le paramètre indiqué ne représente pas un entier, il s'agit d'une exception de type **NumberFormatException** ; on souhaite que le programme affiche par exemple : « L'argument doit être entier », puis qu'il se termine.
3. Si le paramètre est négatif, on souhaite que le programme affiche par exemple : « ExceptionNombreNegatif : -4 est négatif : la factorielle n'est pas définie », puis qu'il se termine. Définissez pour cela votre propre classe d'exception (**ExceptionNombreNegatif**).
4. Si L'argument est un entier trop grand, affichez par exemple : « ExceptionNombreNegatif : 20 est trop grand pour ce programme », puis qu'il se termine Définissez pour cela votre propre classe d'exception (**ExceptionNombreGrand**)

**NB.** : Le plus grand entier peut s'écrire en Java : **Integer.MAX\_VALUE**



## Corrigé

### Exercice 4 (Gestion des Exceptions)

1. S'il n'y a pas de paramètre sur la ligne de commande, il s'agit d'une exception de type **ArrayIndexOutOfBoundsException** ; on souhaite que le programme affiche par exemple : « Indiquez un nombre entier sur la ligne de commande », puis qu'il se termine.

```
...
try
{
    n = Integer.parseInt(argv[0]);
    for (i = 2; i <= n; i++){
        factorielle *= i;
    }
    System.out.println("Voilà la factorielle de " + n + " : " + factorielle);
}
catch (ArrayIndexOutOfBoundsException exc)
{
    System.out.println("Indiquez l'entier dont on souhaite la factorielle " +
        "sur la ligne de commande");
}
}
```

2. Si le paramètre indiqué ne représente pas un entier, il s'agit d'une exception de type **NumberFormatException** ; on souhaite que le programme affiche par exemple : « L'argument doit être entier », puis qu'il se termine.

```
catch(NumberFormatException exc){
    System.out.println("L'argument doit être entier");
}
```

3. Si le paramètre est négatif, on souhaite que le programme affiche par exemple : «ExceptionNombreNegatif : -4 est négatif, la factorielle n'est pas définie », puis qu'il se termine. Définissez pour cela votre propre classe d'exception (**ExceptionNombreNegatif**).

```
...
n = Integer.parseInt(argv[0]);
if (n < 0) throw (new ExceptionNombreNegatif(n));
for (i = 2; i <= n; i++) {
    ...
}
```

```
class ExceptionNombreNegatif extends Exception {
    int valeur;

    ExceptionNombreNegatif(int val) {    valeur = val;    }

    public String toString() {    return super.toString()+ ":"+valeur + " est négatif";    }
}
```



4. Si L'argument est un entier trop grand, affichez par exemple : « ExceptionNombreNegatif : 20 est trop grand pour ce programme », puis qu'il se termine Définissez pour cela votre propre classe d'exception (**ExceptionNombreGrand**)

```
...  
for (i = 2; i <= n; i++){  
    if (factorielle > Integer.MAX_VALUE / i)  
        throw (new ExceptionNombreGrand(n + " est trop grand pour ce  
            programme,\n"));  
    factorielle *= i;  
...  
catch(ExceptionNombreGrand exc){  
    System.out.println(exc);  
}
```

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