

*TP n°5 – Exceptions*

**Exercice 1** Nous reprenons la classe `Main` de l'exercice 3 de la feuille de TP2 dont la méthode `main` calculait la somme de tous les entiers donnés en argument de la ligne de commande. Le code est fourni ci-dessous. On suppose que la liste des arguments peut contenir des valeurs qui ne sont pas des entiers, il faudra donc les ignorer. Afficher le résultat du calcul ainsi que le nombre de valeurs ignorées. On pourra consulter la Javadoc de la méthode `parseInt` pour gérer les valeurs incorrectes de manière adaptée.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int somme =0;  
        int nbErreurs = 0;  
        for (String arg : args){  
            somme += Integer.parseInt(arg);  
        }  
        System.out.println("Somme des arguments " + somme);  
    }  
}
```

**Exercice 2** Écrire une classe `Facto` fournissant deux méthodes :

- **private** static long `factRec (int n)` qui calcule la factorielle d'un entier positif ou nul et retourne un entier de type `long`. On fera donc l'hypothèse que le paramètre fourni sera toujours positif ou nul. Ajouter une pré-condition nécessaire au bon fonctionnement de la méthode sous forme d'une assertion.
- **public** static long `factorielle (int n)` qui calcule la factorielle de l'entier passé en paramètre s'il est positif ou nul et lève une exception dans le cas contraire. Quel type d'exception va-t-on lever ? Ajouter les commentaires destinés au générateur de documentation Javadoc, en particulier l'annotation `@throws`.

Écrire une classe `Main` contenant la méthode `main`. Le programme demandera en boucle à l'utilisateur de saisir un entier dont il souhaite obtenir la factorielle ou bien de saisir 'q' pour quitter. Le programme ne devra pas s'interrompre si l'utilisateur saisit autre chose qu'un entier positif ou nul mais lui demander de ressaisir un entier.

Tester le programme afin d'identifier d'autres valeurs susceptibles de produire un résultat erroné, en particulier sur les valeurs 19, 20 et 21.

Quelle pré-condition doit-on ajouter à la méthode `factRec` pour qu'elle fonctionne correctement sachant qu'elle retourne un entier de type `long`.

Modifier le contrat la méthode publique `factorielle` ainsi que le `main` de la classe `Main`.

**Exercice 3** Reprendre l'exercice 2 du TP3.

Créer un fichier `employes.txt` contenant quelques lignes de la forme `nom salaireBase annee mois jour`. Exemple : Dupont 2000,5 1978 12 1

Le fichier pourra être enregistré dans le projet IntelliJ.

Créer une autre classe `MainBis` contenant le main.

- Créer un objet de type `Personnel`.
- Lire les lignes du fichier avec la classe `Scanner`. On pourra utiliser le constructeur de la classe `Scanner`:

```
public Scanner(File source)  
    throws FileNotFoundException  
  
Scanner sc = new Scanner (new File("employees.txt")) ;
```
- A partir de chaque ligne, créer un employé et l'ajouter au personnel (On suppose que les lignes du fichier ne comportent aucune anomalie).
- Afficher le personnel.