

n	
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40 320
9	362 880

## TP Factorielle (Custom Exception)

# Votre exception métier, vos logs

thèmes : héritage, exception métier, bloc try-catch, framework de logging  
Log4J 2

**Introduction** : A nouveau, nous sommes sur le projet de « La Factorielle » et vous repartez de votre précédent projet « TP Mon premier JavaBean » qui utilise les « long ». Mais cette fois, vous levez les anomalies avec votre exception métier (autre terme : exception personnalisée, custom exception ).

## 1 Ecriture de mon exception métier

Codez votre exception avec son nom : fr.afpa.math.MathException qui dérive directement de java.lang.Exception (checked). Vous codez un constructeur avec un paramètre qui correspond au message expliquant l'erreur. Il y a aussi un constructeur sans paramètre qui force un message générique. En premier, vous testez dans le main.

## 2 Utilisation exception métier

Dans la méthode "factorielle" (récursive), remplacez l'exception prédéfinie IllegalArgumentException par notre exception métier contrôlée : MathException  
Indiquez ici le rajout à la signature de la méthode factorielle() : .....

## 3 Le programme continue malgré la saisie erronée

Votre applicatif (la classe fr.afpa.util.CalculFactor) demande à l'opérateur un entier naturel puis affiche la valeur calculée. Dans la méthode main, vous captez les anomalies MathException. Dans le catch, vous affichez le message explicatif dans une boîte de dialogue showMessageDialog. Le programme ne s'arrête pas.

## 4 L'exception métier remplace une exception de bas niveau

Dans le main, vous captez l'exception de bas-niveau NumberFormatException (sur le parseLong), et substituez par une exception métier MathException et, si possible, continuez le programme . C'est le message contenu dans MathException qui doit distinguer toutes les anomalies entre elles.

## 5 Objet Throwable référencé dans l'exception

Reprenez la question précédente, et attachez l'exception de bas niveau NumberFormatException dans l'exception métier.

Tournez la page SVP 

## 6 Loguez les erreurs avec le framework Log4J 2.x

Pour ceux qui avance vite, consultez de la documentation sur le framework de logging Log4J 2.17.x

Consultez l'excellente page :

[https://koor.fr/Java/Tutorial/java\\_logging\\_log4j2.wp](https://koor.fr/Java/Tutorial/java_logging_log4j2.wp)

Travail à réaliser :

A chaque exception métier que vous levez, en plus, vous loguez une anomalie de niveau « WARNING ».

Vérifiez votre journal.

Au choix, vous développez :

- soit avec un projet dit « classique » c.a.d. avec utilisation des librairies de Log4J2 (downloadez un ZIP sur le site officiel (<https://logging.apache.org/log4j/2.x/>) ou le serveur habituel)
- soit avec un projet mavenisé+dépendance à LOG4J 2.x

Rq : Dans le cas du projet classique, seuls les deux JAR suivants sont nécessaires : log4j-core-2.x.x.jar et log4j-api-2.x.x.jar