

Objectifs :

- ✓ Gestion des Exceptions

Exercice 1 :

La division par zéro crée une `ArithmeticException`.

1. Créez une classe d'objet « ENTIER » qui contient une variable « A » de type `int`. Ecrivez le constructeur de cette classe.
2. Munissez cette classe de la méthode `division`. Cette méthode retourne le résultat (*double* de la division de la variable « A » d'un objet « ENTIER » par la variable « A » d'un autre objet « ENTIER » appelé *diviseur*. L'objet *diviseur* sera passé en paramètre d'entrée de la méthode `division`.
3. Testez votre méthode `division` sur deux entiers quelconques. Recommencez avec un diviseur égal à zéro. Que se passe-t-il ?
4. Dans la méthode `division`: testez le bloc qui effectue l'opération. Si une `ArithmeticException` est générée, affichez le message "Division impossible".

Exercice 2 :

1. Ecrire un code JAVA qui calcule le factoriel d'un entier.
2. Essayer de ne pas faire figurer sur la ligne de commande le paramètre attendu et de mettre un paramètre non-entier. (Une exception est signalée)
3. Modifier le programme pour que, dans chacun des cas, l'erreur soit précisée à l'utilisateur.
4. Développer un mécanisme de propagation de l'exception (*throws ...*)

Exercice 3 :

1. Ecrire une fonction qui prendra en paramètre un entier `n` et renverra une valeur comprise entre 1 et `n`, tapée au clavier par l'utilisateur. Lever et traiter les erreurs rencontrées
Les différentes erreurs qui pourront se produire seront :
 - `n` est inférieur ou égal à 1
 - l'utilisateur a entré un nombre qui n'est pas compris entre 1 et `n`
2. Le nombre entré sera la taille d'un tableau des entiers, ce tableau est d'une référence nulle (`int [] t=null`)
 - Créer une classe qui permet de définir une exception avec un constructeur qui initialise le message d'erreur correspondant.
3. Dans les deux cas précédents, modifier le programme et développer un mécanisme de propagation de l'exception (*throws ...*) en ajoutant les classes nécessaires pour définir toutes les erreurs rencontrées dans le programme.

Exercice 4 :

Pour accéder à la base de données Oracle, il faut saisir le login et le mot de passe système. Notre objectif est de gérer les problèmes d'accès en séparant les exceptions pour une meilleure gestion.

1. Écrire un programme qui demande en boucle un nom d'utilisateur (login) et un mot de passe (pwd) jusqu'à recevoir un login/pwd correct. Le seul utilisateur référencé sera *scott / tiger* (à mettre en constante dans la classe principale).
2. Implémenter les exceptions suivantes:
 - *LoginException* qui se produit lorsque l'utilisateur saisit un login inexistant
 - *PwdException* lorsque le mot de passe est erroné
 - *InputLength* lorsque le login où le pwd saisi dépasse 10 caractères.

Remarque: La lecture de l'entrée standard peut lever une *java.io.IOException...*