

Travaux Dirigés/Pratiques : Les exceptions

EX1 : (Déclenchement et traitement d'une exception)

- Réaliser une classe **EntNat** permettant de manipuler des entiers naturels (positifs ou nuls). Pour l'instant, cette classe disposera simplement :
 1. d'un **constructeur** à un argument de type int qui générera une exception de type **ErrConst** (type classe à définir) lorsque la valeur reçue ne conviendra pas.
 2. d'une méthode **getN** fournissant sous forme d'un int, la valeur encapsulée dans un objet de type EntNat.

Question : Ecrire un petit programme d'utilisation qui traite l'exception **ErrConst** en affichant un message et en interrompant l'exécution.

EX2: (Transmission d'information au gestionnaire)

- Adapter la classe **EntNat** de l'exercice 1 et le programme d'utilisation de manière à disposer dans le gestionnaire d'exception du type **ErrConst** de la valeur fournie à tort au constructeur.

Problème : (Synthèse du chapitre)

- Réaliser une classe permettant de manipuler des entiers naturels (positifs ou nuls) et disposant :
 1. d'un constructeur à un argument de type int ; il générera une exception **ErrConst** si la valeur de son argument est négative.
 2. de méthodes statiques de somme, de différence et de produit de deux naturels ; elles généreront respectivement des exceptions **ErrSom**, **ErrDiff** et **ErrProd** lorsque le résultat ne sera pas représentable ; la limite des valeurs des naturels sera fixée à la plus grande valeur du type int.
 3. Une méthode d'accès **getN** fournissant sous forme d'un int la valeur de l'entier naturel. On s'arrangera pour que toutes les classes exceptions dérivent d'une classe **ErrNat** et pour qu'elles permettent à un éventuel gestionnaire de récupérer les valeurs ayant provoqué l'exception.

Questions : Ecrire deux exemples d'utilisation de la classe :

1. l'un se contentant d'intercepter sans discernement les exceptions de type dérivé de **ErrNat**.
2. l'autre qui explicite la nature de l'exception en affichant les informations disponibles.

Les deux exemples pourront figurer dans deux blocs try d'un même programme.

Remarque : la plus grande valeur entière est définie par : **Integer.MAX_VALUE**