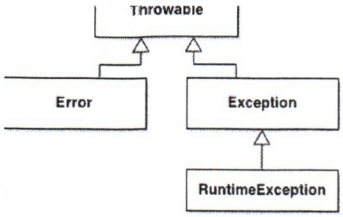

0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40 320
9	362 880

TP Factorielle

Gestion des erreurs avec les exceptions prédéfinies

thèmes : méthode, bloc try-catch, mot réservé throw et throws, instanciation d'exception prédéfinie « unchecked », test unitaire

1 Item: Tests aux limites du programme fourni

Rappel : si vous entrez les valeurs suivantes : 0, 1, 3, -1, 20, 21, 10 000

Quelles sont les valeurs pour lesquelles le résultat est juste ? Expliquez : a<..resultat<21

2 Item : robustesse

Pour la saisie et dans le cas où vous avez utilisé la méthode showInputDialog, vous entrez "azerty". Notez le nom entièrement qualifié de l'exception survenue:
.....NumberFormatException.....(java.lang)
Que pensez-vous de la robustesse de cette application ?
Est-ce une exception checked ou unchecked ?, expliquez : unchecked
.....

3 Item: L'application doit continuer en cas de saisie erronée

Le programme doit continuer malgré les mauvaises saisies de l'opérateur. Dans la méthode main, si possible, essayez de capter les exceptions liées à la saisie. Affichez un message explicatif dans une boîte de dialogue modale de type JOptionPane.showMessageDialog et faites continuer la boucle générale programme.

4 Item: Cas d'erreur IllegalArgumentException

Dans la méthode factorielle et pour les cas d'un argument hors limites, levez l'exception pré-existante "IllegalArgumentException" avec dedans un message explicatif. Pour l'instant, ne captez pas l'exception que vous levée.

5 Item: Le programme continue dans tous les cas d'erreur

Dans la méthode main, captez les anomalies IllegalArgumentException, affichez le message explicatif dans une boîte de dialogue (JOptionPane.showMessageDialog) et continuez le programme. Attention, le catch de IllegalArgumentException doit être placé après le catch de NumberFormatException . Expliquez pourquoi : .....
.....Du..plus..precis au plus large

6 Item : test unitaire (non-JUnit)

Classe TestMath et son main

Pour les cas où la méthode factorielle lève l'exception (ex : 77, -1, 21,…), vous faites des tests unitaires (non-JUnit pour l'instant) dans un main.

Dans l'exemple fourni plus-bas, on capte l'exception attendue avec un bloc try-catch et on continue le programme.

```
//un test unitaire sur la métode factorielle
try {
     Math.factorielle(45);
     System.err.println(" :Test NOK");

} catch (IllegalArgumentException e) {
     System.out.println(" :Test OK :" + e);
} catch (Exception e) {
     System.err.println(" :Test NOK : La factorielle de 45 est
hors limite " );
}
```

7 Item : import static d'une méthode de Math

Dans le test unitaire, vous continuez les « import static » des méthodes (static) de la classe Math.

Quelle est l'utilité? : ……

Annexe : extrait de la hiérarchie des exceptions