



TP factorielleBig

Calcul sur les grands nombres

0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40320
9	362880

Thèmes : méthode, Exception, Error, la classe wrapper

BigInteger, tests JUnit

1 Méthode factorielleBig

Consultez la documentation et les exemples de programmes mis en ligne sur les BigInteger.

Toujours dans la classe fr.afpa.math.Math du projet FrameWorkAfpa, codez une méthode factorielleBig qui calcule la factorielle en utilisant les objets BigInteger.

Ecrivez ici la signature que vous prévoyez :

.....

Votre applicatif (la classe fr.afpa.util.CalculFactorBig) demande à l'opérateur un entier naturel puis affiche la valeur calculée.

Cette fois, pour les nombres supérieurs à 20, comparez le bon résultat avec votre calculatrice sous Windows. Essayez de trouver à partir de quel nombre, on dépasse les capacités de votre application :

En effet ? avec des grands nombres comme paramètre, on atteint aussi les limites du BigInteger. Dans ce cas, vous pouvez capter l'erreur « StackOverflowError » et signalez par un showMessageDialog avec le message : « dépassement de capacité » (comme votre calculatrice).

2 Tests unitaires JUnit (pour ceux qui avancent vite)

Avec les facilités d'Eclipse, complétez le « cas de test » MathTest avec les tests unitaires de la méthode factorielleBig.

a- Vous testez les valeurs: 0 , 1 , 3, 20 , 21, -1, 45, 7000...

Ce qui nous donne les tests unitaires suivants :

```
@Test testFactorielleBigZero() ,
@Test testFactorielleBig1() ...
@Test testFactorielleBigMoinsUn(),
... , etc
```

b- Si elle existe, vous testez une valeur qui dépasse les capacités de votre applicatif (et qui lève une exception)

c- Si vous le souhaitez, vous pouvez utiliser un test paramétré qui remplacera beaucoup de vos tests unitaires précédents

Rappel des recommandations pour JUnit: **une assertion** par @test unitaire et présence **du message explicite en cas d'échec**.



3 Votre calculatrice sur Android

Pour ceux qui avancent vite et en autonomie.

Développez une calculatrice sur Android et implantez-la sur PlayStore

