



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

Méthode static récursive Test unitaire (non-JUnit)

Objets : la récursivité sur une méthode, les tests

unitaires (non-JUnit) dans un main

Contexte : Vous codez une application fr.afpa.util.CalculFactor qui calcule la factorielle d'un nombre saisi par l'opérateur.

1 Question: Le sous-programme factorielle

La factorielle d'un nombre x vaut $x! = x * (x-1) * (x-2) * \dots * 2 * 1$

D'où, on remarque que: $x! = x * (x-1)!$

Autrement dit : $f(x) = x * f(x-1)$

Java est un langage qui accepte l'appel récursif d'une méthode.

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

Dans classe fr.afpa.math.Math du projet FrameworkAfpa, créez une nouvelle méthode statique fr.afpa.math.Math.factorielle qui utilise la **récursivité**.

Si nécessaire, vous avez le pseudo-code dans la page wikipédia "algorithme récursif". Voir aussi la vidéo des teachers du net.

Réfléchissez à la condition qui stoppe l'appel récursif. Quelle est la zone mémoire qui est largement sollicitée par la récursivité ? :

2 Item : tests unitaires de la méthode factorielle

Si nécessaire, vous créez une classe (le cas de test) fr.afpa.math.TestMath (Le sourceFolder est toujours src).

Dans TestMath, vous développez les méthodes-tests-unitaires pour les appels de **factorielle** avec les paramètres suivants: 0, 1, 20, 21, -1, 45

Pour le nom des méthodes de test, vous adoptez avec la convention suivante : factorielleTestZero, factorielleTest20, factorielleTest21, factorielleTest45, factorielleTestMoinsUn, ...

Généralement, il faut un seul test par méthode (une seule assertion par méthode).

Affichez le résultat « test OK » ou « test en échec + explication » (en rouge).

Rq : voir le code de trois tests unitaires au verso

```

                                TestMath.java
package item1_queParseLong_fourniStagiaire;
import static
item1_queParseLong_fourniStagiaire.Math_InitialRecuratif.factorielle;

public class TestMath { //le cas de test //Test Case
public static void main(String[] args) {
    factorielleTestZero();
    factorielleTest2();
    factorielleTestMoinsSept();
}
public static void factorielleTestZero() { //test unitaire
    //assertion : La factorielle de zero vaut 1
    if( factorielle(0) == 1 )
        System.out.println(" :Test OK");
    else
        System.err.println(" :Test NOK : La factorielle de
zero vaut 1");
}
public static void factorielleTest2() {
    //A NE PAS FAIRE !!! plrs assertions dans une test unitaire
    System.out.print("Pas très précis, on ne sait pas quel test
échoué");
    if( factorielle(0)==1 && factorielle(1)==1 &&
factorielle(3)==6)
        System.out.println(" :Test OK");
    else
        System.err.println(" :Test NOK:Le factorielleTest2 à
échoué!");
}
public static void factorielleTestMoinsSept() {
    if( factorielle(-7) == -1 )
        System.out.println(" :Test OK ");
    else
        System.err.println(" :Test NOK : La factorielle de
-7 retourne -1");
}}

```