

Solution factorielle récursive et codes d'erreur

CalculFactorielle_InitialRecuratif.java

```
public class CalculFactorielle_InitialRecuratif
{
/**
 * @return la factorielle de x
 * @return -1 paramètre négatif
 * @return -2 paramètre trop grand
 */
public static long factorielle (long x)
{
    // Tester les cas limites n < 0 , n=22 , et n très grand
    if( x < 0 )
        return -1;          // --> on sort car erreur
    else if( 21 <= x){
        return -2;          // --> on sort car erreur
    }
    else if (x <= 1)
        return 1;           // --> pour 0 ou 1, on sort (arrêt de la
    // récursivité)
    else
        return x * factorielle (x - 1); // Appel récursif à factorielle
}

public static void main (String [] args)
{
    String s = null;
    do
    {
        // Saisie d'un texte avec une boite de dialogue Ok / Annuler

        s = JOptionPane.showInputDialog ("Entrez un nombre :");

        if (s != null) // s est egal a null si l'utilisateur a choisi
        // Annuler
        {
            long n = Long.parseLong (s); // Conversion du texte saisi en
            // nombre
            long factN = factorielle (n); // Calcul de la factorielle
            if( factN < 0 )
                JOptionPane.showMessageDialog (null, "Résultat
            // impossible pour le nombre " + n + " code erreur :" + factN );
            else
                JOptionPane.showMessageDialog (null, n + " ! = " +
            // factN);
        }
    }
    while (s != null);
    System.exit(0);
}
```

Exemple de test JUnit5

```
DEMO2_CalculFactorielleTest_transmis.java
package demo_JUnit5.test;

import static org.junit.Assert.*;
//import static org.junit.Assert.assertEquals;
//import static org.junit.Assert.assertFalse;
//import static org.junit.Assert.assertTrue;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.*;
//import org.junit.jupiter.api.AfterEach;
//import org.junit.jupiter.api.BeforeEach;
//import org.junit.jupiter.api.Test;

import item3_5_mieux.DEMO2_CalculFactorielle;

class DEMO2_CalculFactorielleTest_transmis {
    @BeforeEach
    void setUp() throws Exception {
    }

    @AfterEach
    void tearDown() throws Exception {
    }

    @Test public void testFactorielle1() {
        assertTrue("La factorielle de zero vaut 1",
            DEMO2_CalculFactorielle.factorielle(0) ==
1);
    }

    @Test public void testFactorielle2() {
        assertFalse("La factorielle de 1 vaut 1",
            DEMO2_CalculFactorielle.factorielle(1) !=
1) ;
    }

    @Test public void testFactorielle3() {
        assertEquals("La factorielle de 3 vaut 6",
            DEMO2_CalculFactorielle.factorielle(3), 6)
;
    }

    @Test public void testFactorielle11() {
        assertThrows(IllegalArgumentException.class,
            ()->{DEMO2_CalculFactorielle.factorielle(21);},
            "La factorielle de 21 ne peut pas être calculé" );
    }
}
```

```
public class DEMO2_CalculFactorielle
```

```
{
```

```
    /**
```

```
     * Méthode calculant la factorielle
```

```
     * @param n nombre dont on calcule la factorielle
```

```
     * @return la factorielle de n
```

```
     */
```

```
    public static long factorielle (long n)
```

```
    {
```

```
        // Tester les cas limites n < 0 , n=22 , et n très grand
```

```
        if( n < 0 || 20 < n){
```

```
            throw new IllegalArgumentException("Argument hors limite");
```

```
        }
```

```
        else if( n <= 1 )
```

```
            return 1;
```

```
            // --> on retourne 1 (arrêt de la
```

```
            récursivité)
```

```
        else
```

```
            return n * factorielle (n - 1); // Appel récursif à factorielle
```

```
    }
```

```
    public static void main (String [] args) //throws IOException
```

```
    {
```

```
        String s = null;
```

```
        do
```

```
        {
```

```
            // Saisie d'un texte avec une boîte de dialogue Ok / Annuler
```

```
            s = JOptionPane.showInputDialog ("Entrez un nombre :");
```

```
            if (s != null) // s est egal a null si l'utilisateur a choisi
            Annuler pour sortir
```

```
            {
```

```
                long n=0;
```

```
                long factN=0;
```

```
                try {
```

```
                    n = Long.parseLong (s);
```

```
                    factN = factorielle (n);
```

```
                } catch (NumberFormatException e) {
```

```
                    JOptionPane.showMessageDialog(null, e);
```

```
                    continue;
```

```
                } catch (IllegalArgumentException e) {
```

```
                    JOptionPane.showMessageDialog(null, e);
```

```
                    continue;
```

```
                }
```

```
                JOptionPane.showMessageDialog (null, n + " ! = " + factN);
```

```
            }
```

```
        }
```

```
        while (s != null);
```

```
        System.exit(0);
```

```
    }
```