

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

TP Factorielle

Méthode statique – boucle

thèmes : boucle, méthode et programmation procédurale, package, gestion des cas particuliers, robustesse d'un programme
but : Vous réalisez une application qui calcule la factorielle d'un nombre saisi par l'utilisateur.

Rappels mathématiques

La factorielle d'un nombre entier naturel « x » vaut:

$$f(x) = 1 * 2 * \dots * (x-2) * (x-1) * x$$

$$\text{ou } f(x) = x * (x-1) * (x-2) * \dots * 2 * 1$$

On désigne aussi la factorielle avec le point d'exclamation: « x! »

Rappel sur les cas particuliers de la factorielle:

- cas particulier : $0! = 1$
- cas aux limites : $1! = 1$
- règle : il n'y a pas de factorielle pour les entiers **négatifs**

1 Item : développement de l'applicatif

Avec votre IDE, créez un nouveau projet FrameworkAfpa

L'applicatif est codé dans le main de la classe fr.afpa.util.CalculFactor (à créer).

L'applicatif utilise une méthode factorielleFor qui se situe dans une autre classe d'un autre package fr.afpa.math.Math (voir plus bas))

Pour résumé, les étapes du main sont les suivantes:

1. Le main demande à l'opérateur un entier naturel,
2. Il appelle la méthode factorielleFor
3. puis il affiche la réponse avec la valeur calculée
4. enfin, il demande à l'opérateur s'il veut continuer calculer la factorielle (O/N)

Vérifiez vos résultats en comparant avec ceux de la calculatrice scientifique de votre poste de travail.

Pour la saisie, vous pouvez utiliser la méthode showInputDialog qui ouvre une boîte de dialogue modale. Pour l'affichage du résultat, vous pouvez utiliser la méthode showMessageDialog

2 item: codez la méthode factorielleFor

Dans une classe fr.afpa.math.Math du projet FrameworkAfpa, vous codez une méthode statique fr.afpa.math.Math.**factorielleFor** qui calcule de manière algorithmique (une boucle) la factorielle d'un entier. L'unique paramètre d'entrée est un long et la méthode renvoie un long comme résultat.

Notez ici la signature de la méthode factorielleFor:

.....

Concernant la gestion des anomalies, prenez la convention suivante:

- Si le paramètre passé à `factorielleFor` est négatif, alors vous renvoyez -1 qui signifie que la méthode ne peut pas calculer la factorielle.

3 item: dépassement des capacités avec le type long

Vérifiez vos résultats en comparant avec ceux de la calculatrice scientifique de votre bureau. Avec des saisies successives, vous notez ici la valeur du paramètre de `factorielleFor` où le résultat est faux (on dépasse la capacité du primitif long) : ...21..... (Pour ces cas, on devra renvoyer la valeur -2)

Donc, concernant la gestion des erreurs, et dans le cas d'un paramètre trop grand, vous prenez la deuxième convention suivante:

- si le paramètre passé à `factorielleFor` est trop grand, alors vous renvoyez -2.

4 Item : Import static d'une méthode de Math

Vous essayez un « import static » de la méthode `factorielleFor` de la classe `Math`.

Quelle est l'utilité de l'import static? :