

# Aide – Métadonnées des méthodes de test & explication métier (JUnit, Maven)

## CalculatorLiteTest

Méthode de test	Ce qui est vérifié	Jeux d'essai (exemples)	Intention métier
<code>add shouldSumTwoNumbers()</code>	L'addition retourne la somme correcte, y compris avec des entiers négatifs.	<code>add(3,4) → 7 ;</code> <code>add(2,-3) → -1</code>	Garantir que l'addition fonctionne en cas de mélange de signes.
<code>subtract shouldSubtractTwoNumbers()</code>	La soustraction calcule <code>a - b</code> correctement, y compris avec <code>b</code> négatif.	<code>subtract(3,4) → -1</code> <code>; subtract(2,-3) → 5</code>	Assurer la cohérence du calcul directionnel (ordre des opérandes).
<code>multiply shouldMultiplyTwoNumbers()</code>	La multiplication gère signes et zéro.	<code>multiply(3,4) → 12</code> <code>; multiply(2,-3) → -6 ;</code> <code>multiply(123,0) → 0</code>	Vérifier le produit, la propagation du zéro et les signes.
<code>divide shouldDivideTwoNumbers integerDivision()</code>	<b>Division entière</b> (tronquée) et gestion des signes.	<code>divide(9,4) → 2 ;</code> <code>divide(9,-3) → -3</code>	Confirmer la sémantique de division <i>entière</i> Java (pas de décimales).
<code>divide byZero shouldThrow()</code>	La division par zéro déclenche une exception.	<code>divide(1,0) → IllegalArgumentException</code>	Sécurité : interdire les opérations non définies.

## StringHelperTest

Méthode de test	Ce qui est vérifié	Jeux d'essai (exemples)	Intention métier
<code>isBlank shouldDetectNullEmptyAndSpaces()</code>	Détection des chaînes <i>blanches</i> ( <code>null</code> , vide, espaces).	<code>null → true ; "" → true</code> <code>; " " → true ; " a " → false</code>	Normaliser les entrées utilisateur vides/blanches.

Méthode de test	Ce qui est vérifié	Jeux d'essai (exemples)	Intention métier
<code>reverse shouldReverseOr ReturnNull()</code>	Inversion de chaîne, <code>null</code> reste <code>null</code> .	<code>null</code> → <code>null</code> ; <code>"abc"</code> → <code>"cba"</code> ; <code>"aba"</code> → <code>"aba"</code>	Outil de manipulation de texte simple et robuste.
<code>isPalindrom e shouldIgnoreCaseAndSpaces()</code>	Palindrome <b>insensible</b> à la casse et aux espaces.	<code>"Kayak"</code> → <code>true</code> ; <code>"n u r s e s r u n"</code> → <code>true</code> ; <code>null</code> → <code>false</code> ; <code>"Test"</code> → <code>false</code>	Validation sémantique de mots/phrases indépendamment du formatage.
<code>repeat shouldRepeatStringNtimes()</code>	Répétition <code>n</code> fois ; <code>String.valueOf(null)</code> attendu pour <code>null</code> .	<code>("X",0)</code> → <code>" "</code> ; <code>("a",3)</code> → <code>"aaa"</code> ; <code>(null,2)</code> → <code>"nullnull"</code>	Génération de motifs/chaînes, même avec <code>null</code> .
<code>repeat negative shouldThrow()</code>	Interdit <code>n &lt; 0</code> par exception.	<code>("x",-1)</code> → <code>IllegalArgumentException</code>	Sécurité des paramètres d'API.
<code>capitalize shouldHandleCases()</code>	Mise en majuscule du 1er caractère, le reste en minuscule ; cas limites.	<code>null</code> → <code>null</code> ; <code>" "</code> → <code>" "</code> ; <code>"hELLO"</code> → <code>"Hello"</code> ; <code>"a"</code> → <code>"A"</code>	Normalisation d'affichage (titres, noms, etc.).
<code>splitWords shouldSplitOnSpaces()</code>	Découpage par espaces successifs ; résultats vides pour <code>null</code> /blanc.	<code>null</code> → <code>[]</code> ; <code>" "</code> → <code>[]</code> ; <code>"a bb ccc "</code> → <code>["a","bb","ccc"]</code>	Tokenisation simple pour traitement de texte.

## RectangleTest

Méthode de test	Ce qui est vérifié	Jeux d'essai (exemples)	Intention métier
<code>constructor shouldRejectNegative()</code>	Le constructeur <b>refuse</b> largeur/hauteur négatives.	<code>(-1,2)</code> et <code>(2,-1)</code> → <code>IllegalArgumentException</code>	Intégrité des données géométriques à l'entrée.
<code>area and perimeter shouldBeCorrect()</code>	Aire et périmètre corrects (avec tolérance <code>1e-9</code> ).	<code>(3,4)</code> → <code>area=12.0</code> ; <code>perimeter=14.0</code>	Fiabilité des calculs de base.
<code>scale negative_s houldThrow()</code>	Facteur de mise à l'échelle négatif <b>interdit</b> .	<code>scale(-0.1)</code> → <code>IllegalArgumentException</code>	Prévenir des transformations non valides.