**TESTS UNITAIRES JUNIT**

Table des matières

[Chapitre 1 2](#_Toc91001280)

[Titre 1 2](#_Toc91001281)

[Titre 2 2](#_Toc91001282)

[Chapitre 2 2](#_Toc91001283)

[Titre 1 2](#_Toc91001284)

[Chapitre 3 2](#_Toc91001285)

[Titre 1 2](#_Toc91001286)

[Chapitre 4 2](#_Toc91001287)

[Titre 1 2](#_Toc91001288)

[Chapitre 5 2](#_Toc91001289)

[Titre 1 2](#_Toc91001290)

[Chapitre 6 2](#_Toc91001291)

[Titre 1 2](#_Toc91001292)

# 

# Définitions des tests unitaires :

Les tests unitaires sont des tests automatisés à l’échelle du développement. Ce n’est pas des tests fonctionnels. Chaque méthode doit être valide pour toutes les instances du problème.

***Exemple :***

Soit p () est une méthode Java, fonction ou un programme

On dit que p() est correcte si :

* Ɐ (x,y) ϵ R2  P() est correcte. => pour chaque instance de de x et y p() est correcte.

Le framework de test unitaire pour Java est Junit

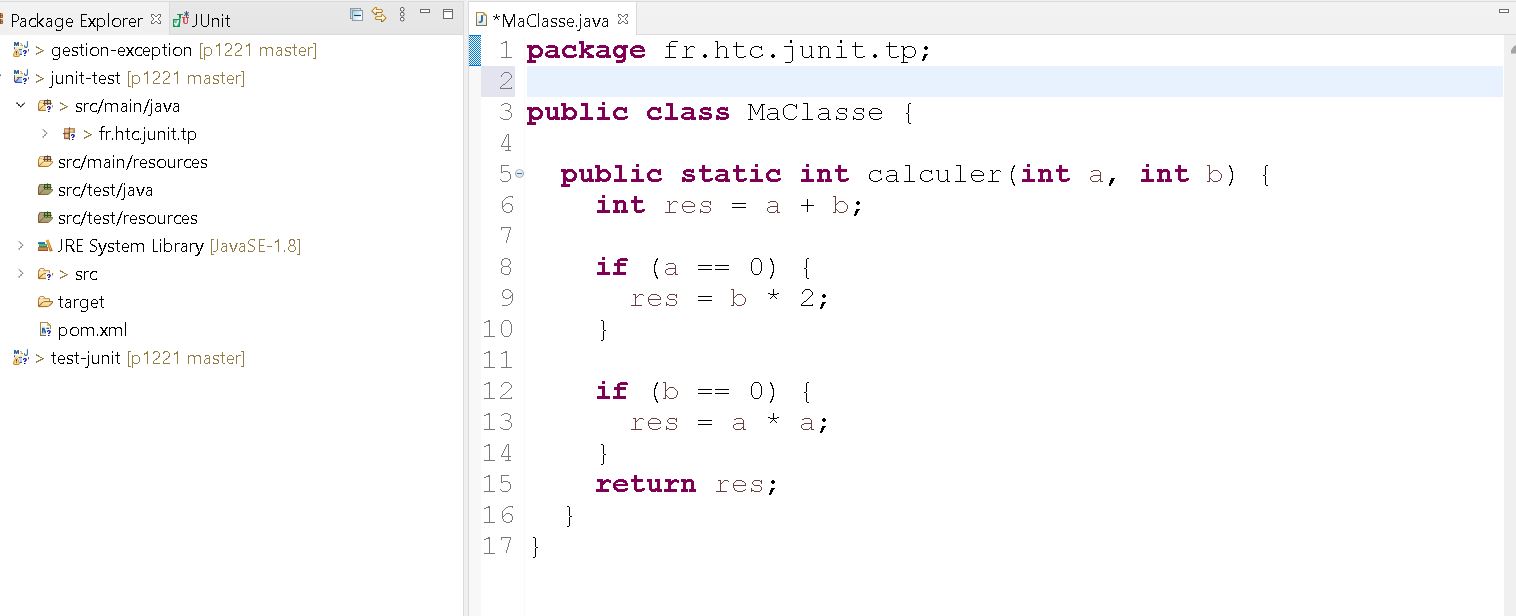
# Junit Définition :

JUnit est un framework open source pour le développement et l'exécution de tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non-régression.

* JUnit propose :
* Un framework pour le développement des tests unitaires reposant sur des assertions qui testent les résultats attendus
* Des applications pour permettre l'exécution des tests et afficher les résultats
* Cela permet de séparer le code de la classe, du code qui permet de la tester.
* Souvent pour tester une classe, il est facile de créer une méthode main() qui va contenir les traitements de tests. L'inconvénient est que ce code "superflu" est inclus dans la classe. De plus, son exécution doit se faire manuellement.
* Les cas de tests sont regroupés dans des classes Java qui contiennent une ou plusieurs méthodes de tests. Les cas de tests peuvent être exécutés individuellement ou sous la forme de suites de tests.

# Exemple simple de Junit :

1. Création d’un projet Maven



1. Écrire une classe qui va contenir les différents tests à réaliser par JUnit. L'exemple est volontairement simpliste en ne définissant qu'un seul cas de tests.



# 

## Titre 1

# 

## Titre 1

# 

## Titre 1