

JUnit 5

1. Introduction à JUnit 5

JUnit est un framework open source pour développer et exécuter de tests unitaires automatisables.

JUnit5 est la dernière version de JUnit. Il y a également d'autres versions de JUnit qui sont encore utilisable et notamment JUnit 3 et JUnit 4. Il y a une rupture de compatibilité entre ces versions de JUnit. Par exemple, l'API JUnit 3 a été placée dans le package **junit.framework** tandis que l'API version 4 a été placée dans le package **org.junit**. La version 5 de l'API JUnit est dans le package **org.junit.jupiter**.

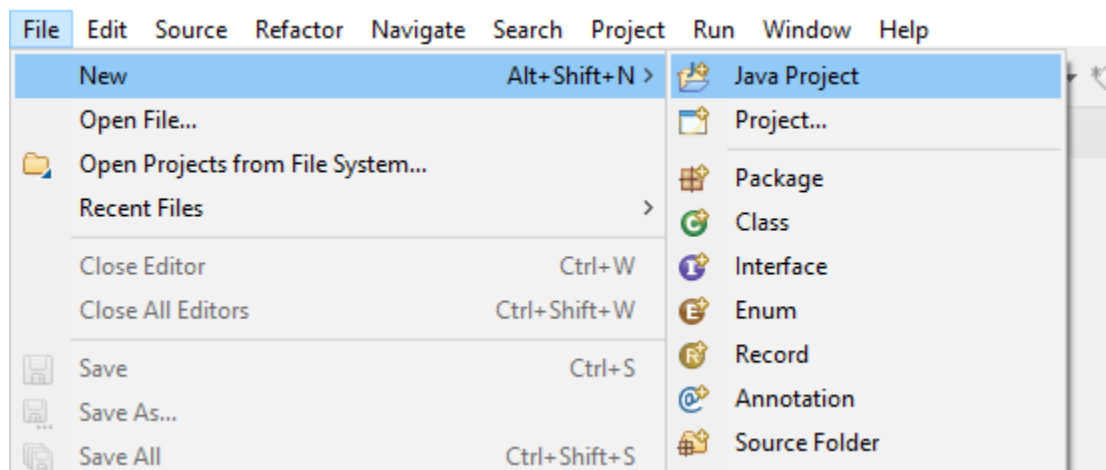
JUnit5 repose en grande partie sur l'utilisation d'un jeu d'annotations. Mais également, elle exploite les innovations apportées par Java 8 et notamment les lambdas et les annotations répétées.

2. 1^{er} projet JAVA avec le support JUnit 5.x

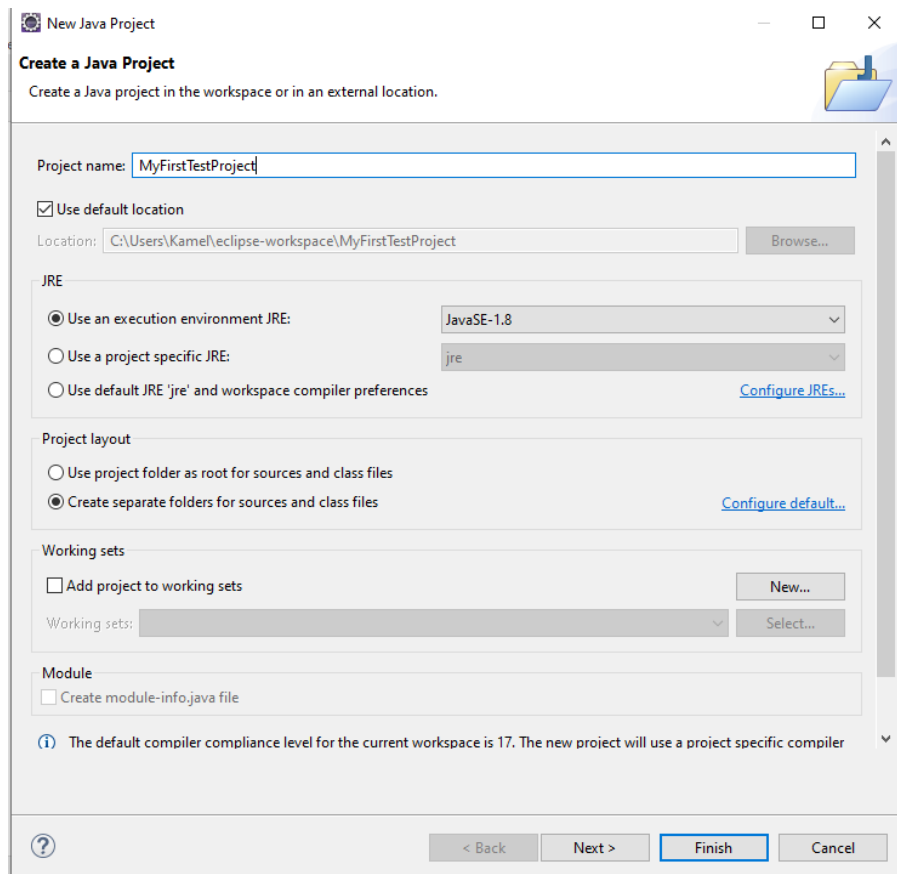
a) On va créer un nouveau projet comme suit:

- Créer un projet avec le nom *MyFirstTestProject*

File → New → Java Project



Tapez le nom du projet *MyFirstTestProject* et cliquez sur *finish*



- On va créer la classe suivante qui s'appelle *MyClass*:

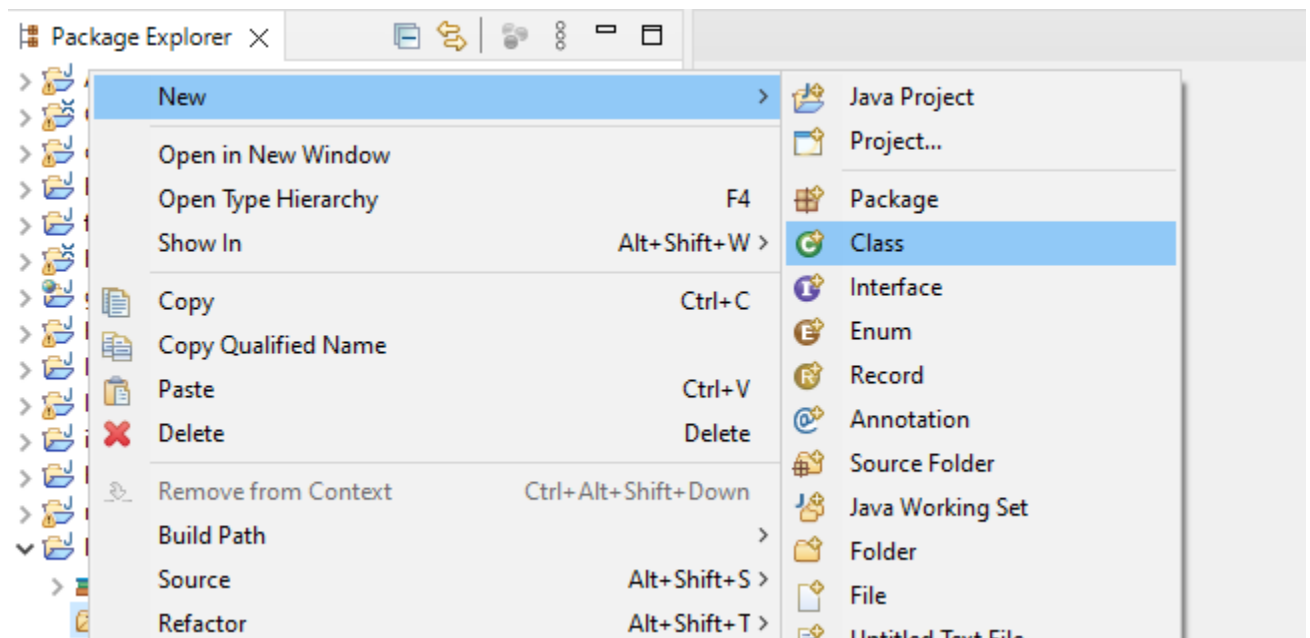
```
public class MyClass{

    public static int calculer(int a, int b) {
        int res = a + b;

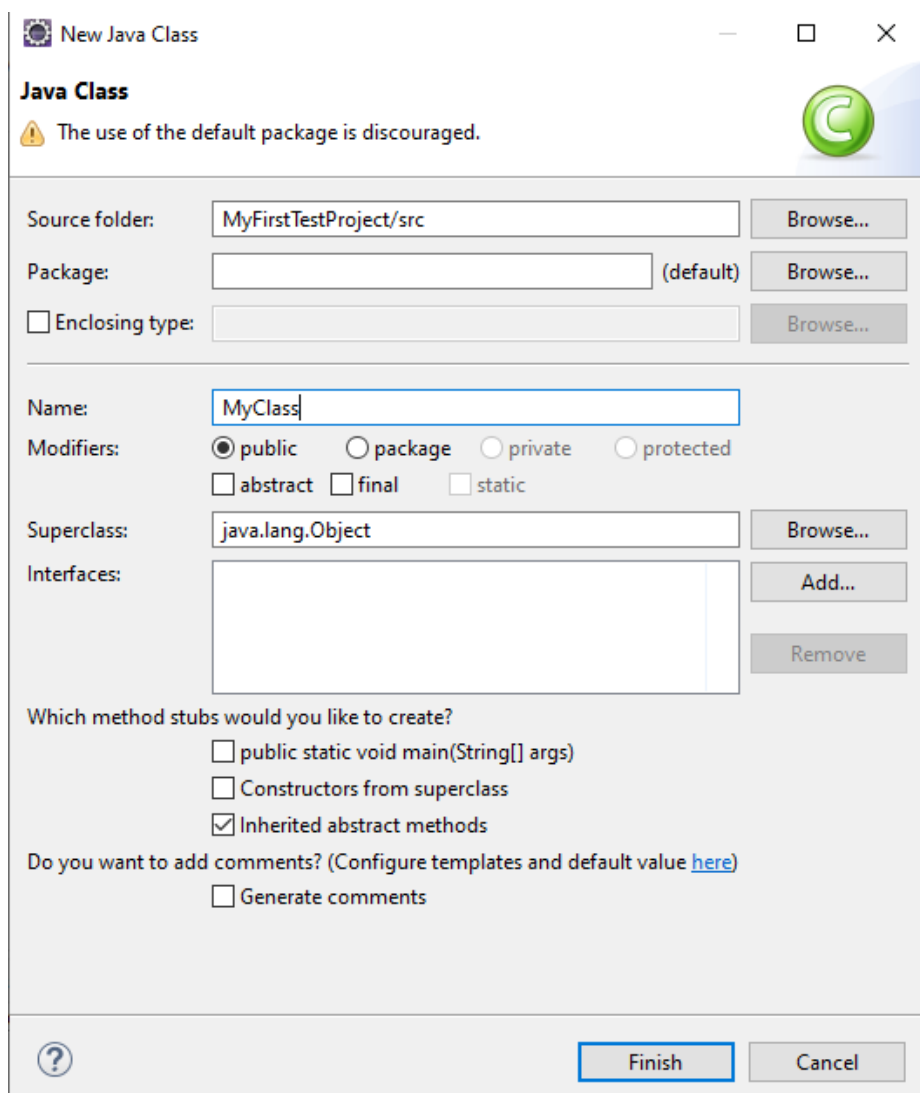
        if (a == 0){
            res = b * 2;
        }

        if (b == 0) {
            res = a * a;
        }
        return res;
    }
}
```

Faites un clic droit sur le projet *MyFirstTestProject* puis *New* → *Classe*



Tapez le nom de la classe *MyClasse* et cliquez sur finish

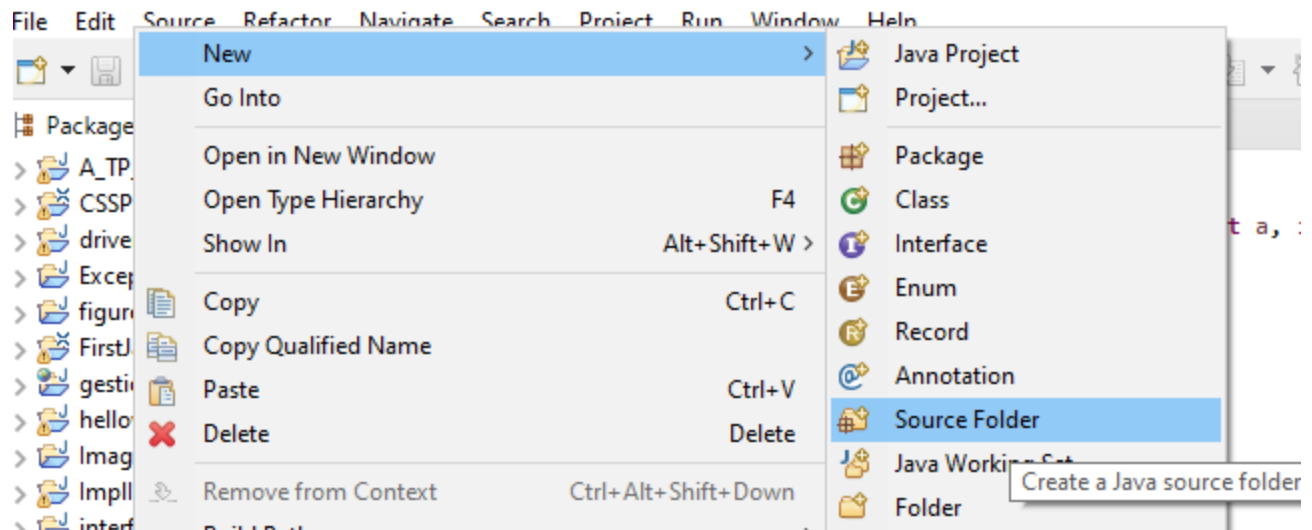


b) On va créer un projet test comme suit:

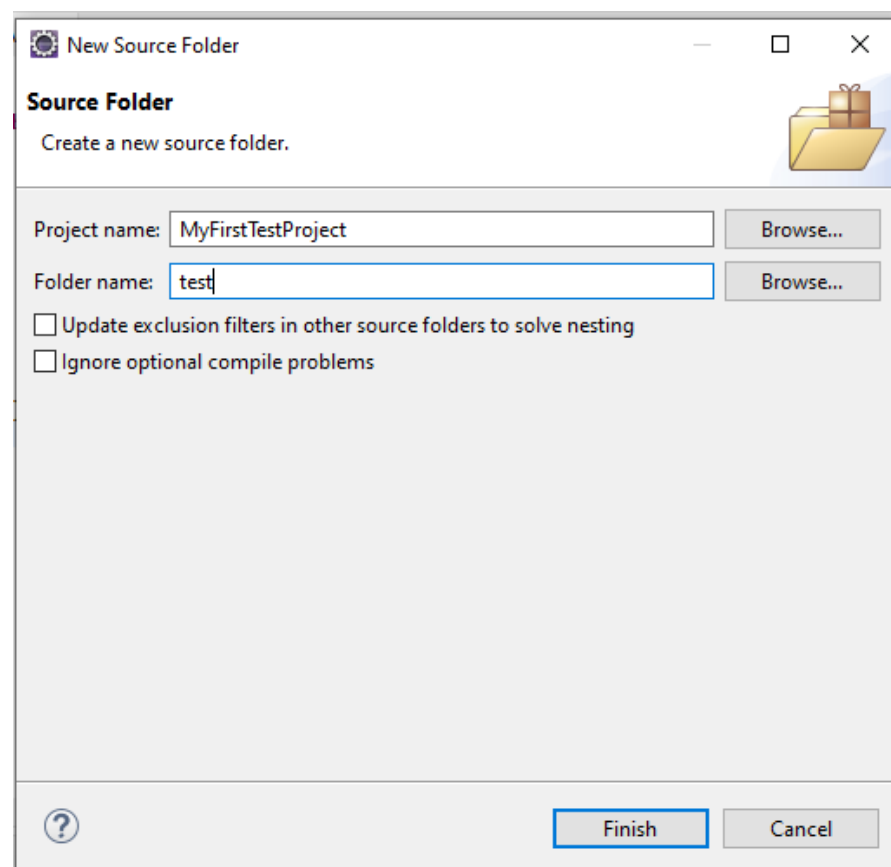
- Créer un dossier **Source Folder** :

Le mélange des codes applicatifs et les codes de tests, dans le même endroit n'est pas une bonne pratique. Pour cela, Eclipse propose la notion de « Source Folder » pour éviter ce mélange. Le dossier « **Source Folder** » contient, des codes sources exactement comme le dossier « **src** ».

Pour créer un dossier Source Folder, cliquez avec le bouton droit de la souris sur le projet puis sélectionnez « **New → Source Folder** ».

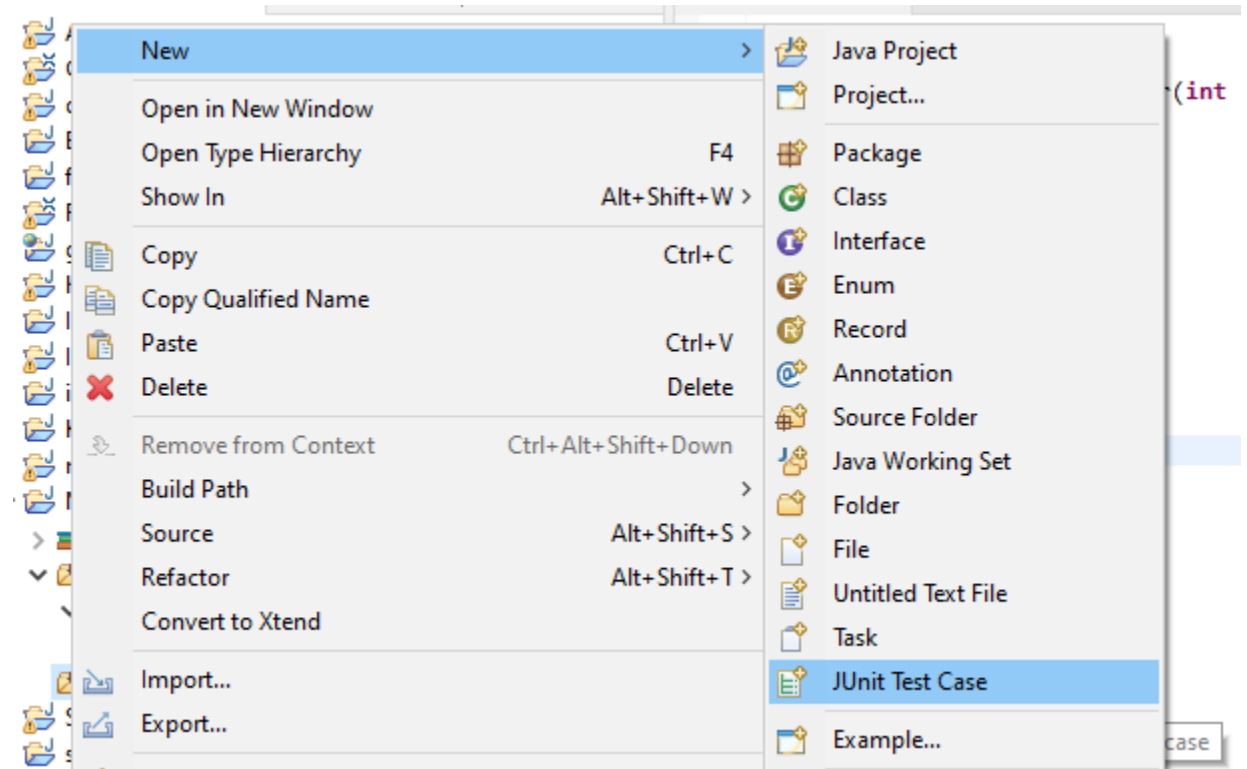


Appellez ce dossier **test** et cliquez sur **Finish**

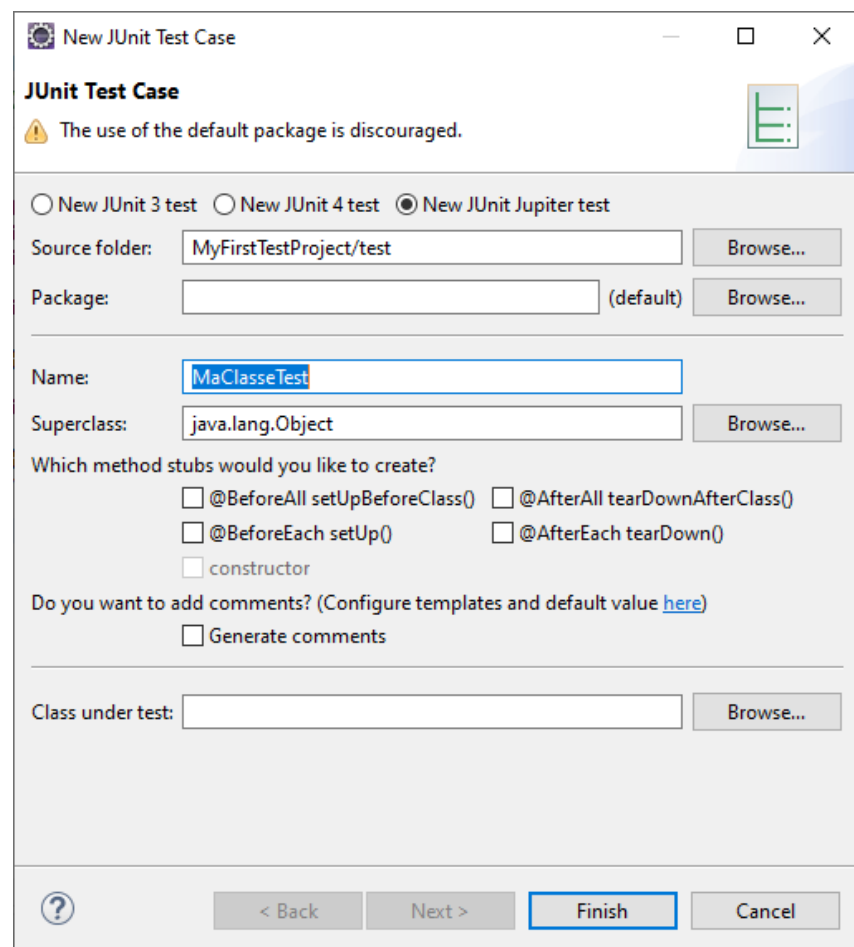


- Créer une classe ***JUnitTestCase***

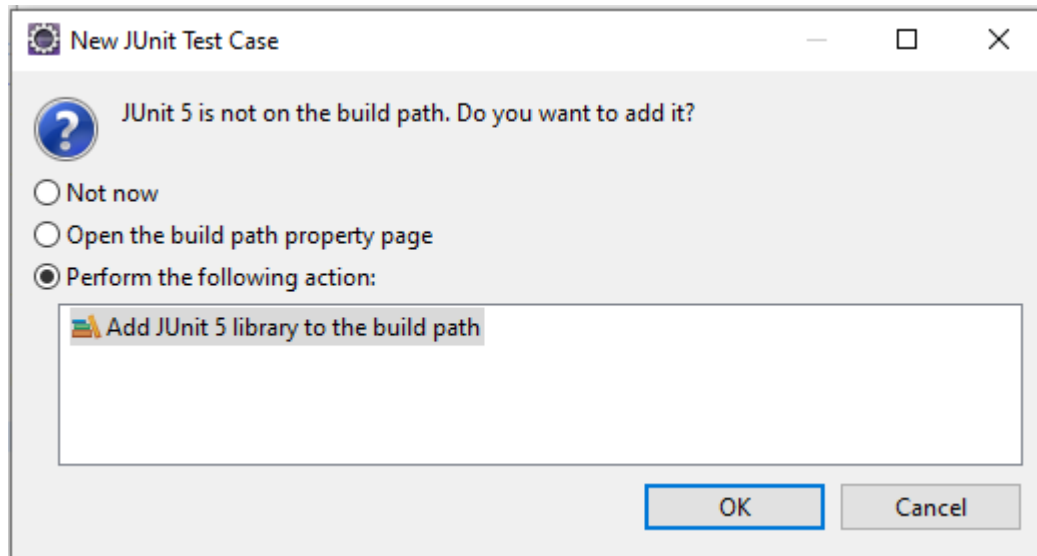
Faites un clic droit sur le dossier ***test*** puis ***New*** → ***JUnitTestCase***



Appellez cette classe de test ***MaClasseTest*** et cliquez sur ***Finish***



- La première fois qu'on ajoute une classe de test à un projet, *Eclipse* propose d'ajouter automatiquement la bibliothèque *JUnit* au *ClassPath*. Acceptez, bien sûr, cette proposition.



- voici à quoi ressembler votre classe de test.

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

class MaClasseTest {
    @Test
    void test() {
        fail("Not yet implemented");
    }
}
```

c) Définir la première méthode de test

La méthode de test doit être **publique**, et qu'elle n'accepte **ni paramètres ni valeur de retour (void)**. Elle est toujours précédée de l'annotation **@Test** (celle du package *org.junit.jupiter.api*).

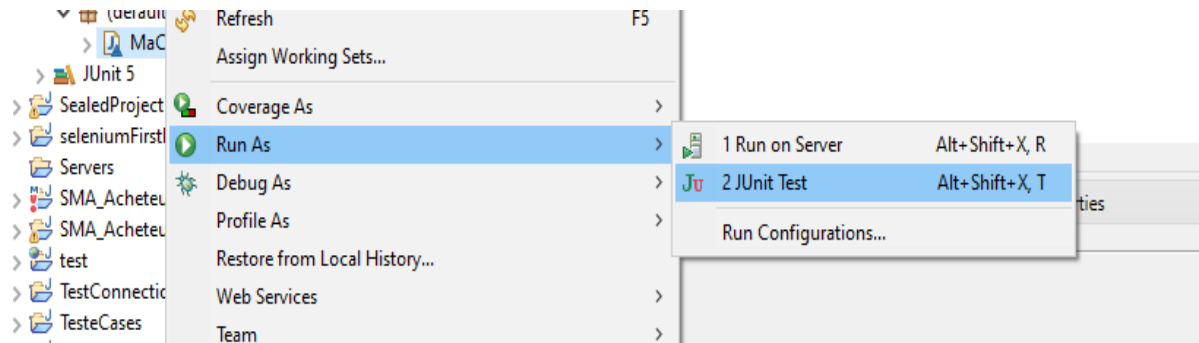
On va définir la méthode test1 comme suit :

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

class MaClasseTest {
    @Test
    public void test1() {
        assertEquals(2, MyClass.calculer(1,1));
    }
}
```

assertEquals est une méthode statique utilisée pour vérifier l'égalité de deux valeurs : le résultat attendu (2 dans cet exemple) et le résultat obtenu (dans ce cas c'est le résultat de la méthode `MyClass.calculer(1,1)`). Le test sera réussi s'il y a une égalité entre le résultat attendu et le résultat obtenu et sera échoué s'il y a une différence.

- Exécuter les tests JUnit



Pour exécuter l'ensemble de test, faire un clic droit sur le nom de la classe ou le package et choisir l'assistant "Run as / JUnit Test". Les résultats du test sont affichés dans la vue "JUnit" comme suit.

