

TP1 - Tests Unitaires avec JUnit 5

1. Présentation générale

L'objectif de ce TP est de vous familiariser avec les tests unitaires en Java en utilisant JUnit 5. Vous apprendrez à créer et exécuter des tests, à utiliser des assertions et à organiser votre code de manière à faciliter les tests.

2. Préparation de l'environnement

Nous allons créer un projet Maven sous Eclipse et y ajouter la dépendance JUnit 5.

2.1. Création d'un projet Maven

1. Lancer Eclipse et créer un nouveau projet Maven :
 - a. File > New > Maven Project
 - b. Cocher "Create a simple project" > Next
 - c. Remplir les informations :
 - i. **Group ID** : edu.uamm
 - ii. **Artifact ID** : tp-junit5
 - iii. **Version** : 1.0
 - iv. **Description** : Projet de test avec JUnit 5
 - d. Valider et générer le projet.

2.2. Ajout de la dépendance JUnit 5

Ouvrir le fichier pom.xml et ajouter la dépendance suivante :

```
<dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.9.2</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.9.2</version>
```

```
</dependency>
</dependencies>
```

Mettre à jour le projet Maven (Maven > Update Project).

3. Implémentation et tests unitaires

3.1. Création de la classe Calculator

Dans src/main/java/edu/uamm/tp, créer une classe Calculator :

```
package edu.uamm.tp;

public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
    public int subtract(int a, int b) {
        return a - b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
    public double divide(int a, int b) {
        if (b == 0) {
            throw new IllegalArgumentException("Division par zéro non permise");
        }
        return (double) a / b;
    }
}
```

3.2. Création des tests unitaires

Dans src/test/java/edu/uamm/tp, créer la classe CalculatorTest :

```
package edu.uamm.tp;

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

```

import org.junit.jupiter.api.Test;

class CalculatorTest {

    private final Calculator calculator = new Calculator();

    @Test
    void testAddition() {
        assertEquals(5, calculator.add(2, 3));
    }

    @Test
    void testSoustraction() {
        assertEquals(2, calculator.subtract(5, 3));
    }

    @Test
    void testMultiplication() {
        assertEquals(15, calculator.multiply(3, 5));
    }

    @Test
    void testDivision() {
        assertEquals(2.0, calculator.divide(6, 3));
    }

    @Test
    void testDivisionParZero() {
        assertThrows(IllegalArgumentException.class, () -> calculator.divide(5, 0));
    }
}

```

4. Exécution des tests

4.1. Avec Eclipse

- Clic droit sur la classe CalculatorTest > Run As > JUnit Test.
- Tous les tests doivent passer avec un indicateur vert.

4.2. Avec Maven

Dans un terminal, naviguer jusqu'au projet et exécuter :

```
mvn test
```

Les résultats des tests s'affichent dans la console.

5. Travail à faire

1. Ajouter une méthode modulo(int a, int b) à Calculator.
2. Ajouter des tests unitaires pour cette nouvelle méthode.
3. Gérer le cas où $b == 0$ et tester ce cas.

6. Rendu du TP

Compresser le projet et le soumettre sous la forme d'une archive tp-junit5.zip sur la plateforme de rendu.

Deadline :