

TD8 : Tests

Objectif du TD

À la fin de ce TD, vous devriez :

1. Connaître les bases de JUnit,
2. Être capable de tester un projet Maven.

Exercice 1 : JUnit

1. Récupérez le projet du TD6 sur Moodle.

Pour rappel, il constitue un jeu interactif simple. Aucun test n'y est implémenté. Nous allons ajouter les tests suivants :

- s'assurer que le nombre choisi par le jeu est bien entre 1 et 100,
- s'assurer que la comparaison entre le nombre entré par le joueur et le nombre choisi est correcte,
- s'assurer que le jeu s'arrête au bout de 10 essais.

Pour cela, nous utiliserons JUnit (framework de test unitaire pour Java).

Les informations suivantes pourront être utiles :

- Les balises `@BeforeEach`, `@BeforeAll`, `@Test`, `@AfterAll`, `@AfterEach` permettent de préciser si les méthodes sont des tests et quand elles doivent être jouées.
- Voici les signatures des méthodes `assertEquals` et `assertTrue` :
 - `static void assertEquals(Object expected, Object actual)`
 - `static void assertTrue(boolean condition)`
 - `static void assertFalse(boolean condition)`
 - `static void assertNotNull(Object object)`

Dans les deux cas, un argument optionnel est possible : une chaîne de caractères pour un message personnalisé.

- Pour accéder à aux attributs privés d'une classe, la *réflexion* et les champs (*Field*) peuvent être utiles. Un *Field* représente une variable d'une instance d'une classe. On peut alors l'obtenir et le manipuler via la réflexion.
 - `Field champ = getDeclaredField("nomDuChamp")` accède au champ privé (si le champ n'existe pas, `NoSuchFieldException` est levée)
 - `champ.setAccessible(true)` permet de le rendre accessible (sans cette ligne, `IllegalAccessException` est levée)
 - `champ.get(objet)` permet de le lire
 - `champ.set(objet, nouvelleValeur)` change sa valeur

2. Créez l'arborescence et le fichier nécessaire aux tests.

3. Suivez l'approche AAA (Arrange - Act - Assert) pour tester les différents points. N'oubliez pas que les tests doivent être indépendants.

Exercice 2 : Maven

1. Modifier le fichier `pom.xml` de sorte que la commande `mvn test` lance les tests du projet.
 - Devrons apparaître dans les dépendances les artefacts `junit-jupiter-api` et `junit-jupiter-engine` du groupe `org.junit.jupiter` (version 5.9.2). Elles ne seront nécessaires que lors des tests (*scope*).
 - Lors du *build* du projet, le *plugin* `maven-surefire-plugin` du groupe `org.apache.maven.plugins` (version 3.0.0-M7) sera nécessaire.