

Qu'est-ce qu'un test unitaire en Java ?

Un test unitaire est un test automatique qui vérifie le comportement d'une unité de code spécifique (comme une méthode ou une classe) dans différents cas d'usage.

Quelles sont les trois étapes d'un test unitaire ?

1. Arrangement : mise en place des conditions nécessaires au test.
2. Action : exécution du code à tester.
3. Affirmation : vérification que le résultat est correct.

Comment structurer une classe de test avec JUnit ?

Créer une classe nommée comme la classe à tester, suivie de 'Test' (ex : AppTest pour App).

Placer cette classe dans le même package, sous le dossier test du projet.

Quelle annotation utiliser pour une méthode de test ?

Utiliser l'annotation `@Test` pour indiquer qu'une méthode est un cas de test.

Comment nommer une méthode de test de manière descriptive ?

Utiliser la convention 'étant donné - quand - alors', ex :

wordCount_whenMultipleWords_returnsRightCount.

Comment utiliser assertEquals dans un test ?

@Test

```
void wordCount_whenMultipleWords_returnsRightCount() {
```

```
    // Given
```

```
    String input = "bonjour le monde";
```

```
    // When
```

```
    int result = App.countWords(input);
```

```
    // Then
```

```
    assertEquals(3, result);
```

```
}
```

Qu'est-ce que la couverture de test ?

La couverture de test mesure la proportion de code exécutée lors des tests.

Elle permet d'évaluer l'efficacité des tests en identifiant les parties du code non testées.

Comment exécuter les tests avec couverture dans IntelliJ IDEA ?

Utiliser l'option 'Run with coverage' (icône en forme de bouclier) pour exécuter les tests et obtenir la couverture du code.