

Un code propre exige des tests propres... même en Java !

Un exemple concret pour illustrer les bonnes pratiques :

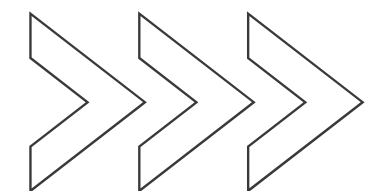
❌ Exemple de test à éviter :



Java ▾



```
1 @Test
2 void test1() {
3     BankAccount a = new BankAccount("user123", 100);
4     a.withdraw(150);
5     assertTrue(a.getBalance() < 0);
6 }
7
8
```



✓ Exemple de test correct

```
Java ▾ ⓘ  
1 @Test  
2 void shouldIncreaseBalance_WhenDepositIsMade() {  
3     // Arrange  
4     BankAccount account = new BankAccount("Lucie Martin", 100.00);  
5     double depositAmount = 50.00;  
6  
7     // Act  
8     account.deposit(depositAmount);  
9  
10    // Assert  
11    assertThat(account.getBalance())  
12        .as("Le solde après dépôt")  
13        .isEqualTo(150.00);  
14 }  
15
```

Pourquoi c'est propre ?

- AAA respecté :
 - Arrange : Un compte avec un solde (100€).
 - Act : Dépôt de 50€.
 - Assert : Vérification du solde final (150€).
- Nom clair : Le scénario est compréhensible sans lire le code.
- Données réalistes : Montants précis (50.00, 100.00), nom explicite ("Lucie Martin").
- Assertion parlante : AssertJ (assertThat) rend le test lisible comme une phrase.

