**Unit Testing**

Mockito is a popular mocking framework used for unit testing Java applications. It allows developers to create mock objects for dependencies.

Spring Boot applications often have layers (Controller → Service → Repository). When testing one layer, you usually want to mock the others to avoid hitting the database.

* @Mock: Creates a mock instance.
* @InjectMocks: Injects mock dependencies into the class under test.
* @ExtendWith(MockitoExtension.class): Enables Mockito in JUnit 5.

**JUnit 5:** JUnit 5 is the latest version of the popular Java testing framework used for writing and running unit tests. 3 components of JUnit5:

1. **JUnit Platform:** Foundation for launching testing frameworks on JVM.
2. **JUnit Jupiter:** This is where you write your tests using annotations like @Test, @BeforeEach, @AfterEach, etc.
3. **JUnit Vintage:** Allows running JUnit 3 and JUnit 4 tests on the JUnit 5 platform.

**What is Unit Testing?**

Unit Testing is a software testing technique here units are tested. A unit is typically smallest testable part of an application, such as a method or class.

A screenshot of a computer program

AI-generated content may be incorrect.

**What is Test Driven Development?**

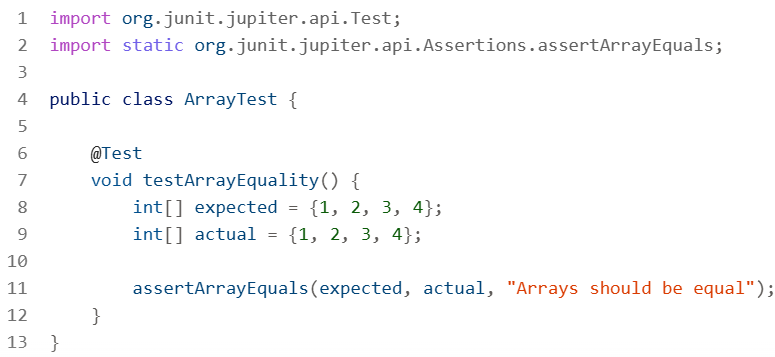
Test-Driven Development (TDD) is a software development approach where tests are written before the actual code.

**A white text on a white background

AI-generated content may be incorrect.**

**@Test Annotation:** It marks a method as test method, and that method will be executed by the testing framework. The method must be public and void in JUnit 4, but in JUnit 5, it can be package-private.

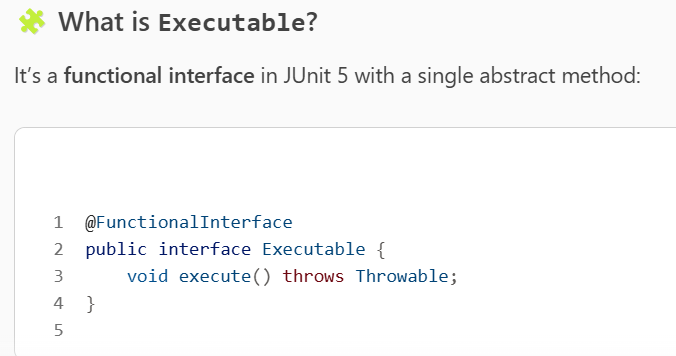
**assertArrayEquals:**

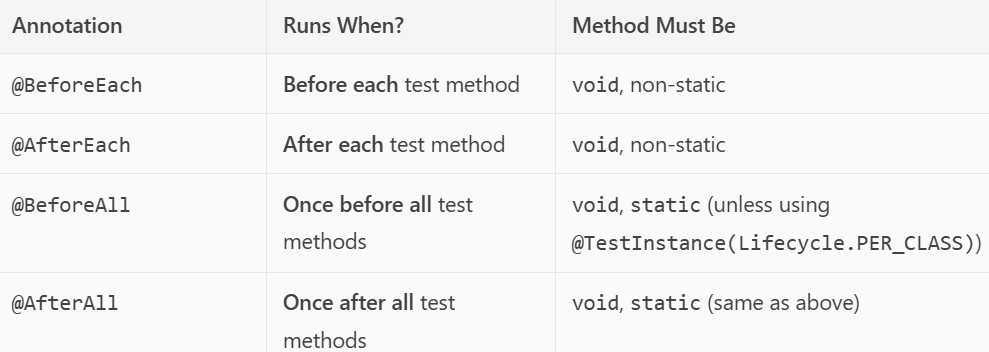


**assertThrows:** used to verify that a specific exception is thrown.

**A screen shot of a computer code

AI-generated content may be incorrect.**

****

****

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a test

AI-generated content may be incorrect.**