**Spring framework**

# **8. Unitest**

# **1. Tìm hiểu các thư viện viết unit test trong dự án java**

Trong các dự án Java, việc viết Unit Test là rất quan trọng để đảm bảo chất lượng và độ tin cậy của mã nguồn. Có nhiều thư viện và framework hỗ trợ viết Unit Test, mỗi thư viện có ưu và nhược điểm riêng. Dưới đây là một số thư viện phổ biến nhất để viết Unit Test trong Java:

### 1. JUnit

* Mô tả: JUnit là một trong những thư viện Unit Test phổ biến và lâu đời nhất trong Java. Phiên bản mới nhất là JUnit 5 (Jupiter), mang lại nhiều cải tiến so với các phiên bản trước.
* Tính năng:
  + Cung cấp các assertions như assertEquals(), assertTrue(), assertNotNull(), v.v.
  + Hỗ trợ các annotations như @Test, @BeforeEach, @AfterEach, @BeforeAll, @AfterAll.
  + Tích hợp với các công cụ xây dựng dự án như Maven và Gradle.
  + Hỗ trợ mở rộng với các test suite và tích hợp với nhiều IDE (Eclipse, IntelliJ IDEA, v.v.)

### 2. Mockito

* Mô tả: Mockito là một framework phổ biến để tạo mock objects trong quá trình viết Unit Test. Mockito giúp giả lập hành vi của các đối tượng phụ thuộc mà không cần phải thực hiện logic thực tế của chúng.
* Tính năng:
  + Giả lập (mock) các đối tượng và xác định hành vi của chúng khi được gọi.
  + Hỗ trợ kiểm tra tương tác giữa các đối tượng (interaction testing).
  + Có thể kết hợp với JUnit để tăng khả năng kiểm thử đơn vị.
  + Hỗ trợ annotation như @Mock, @InjectMocks.

### 3. AssertJ

* Mô tả: AssertJ cung cấp một tập hợp phong phú các assertion mượt mà, giúp cho mã kiểm tra (test code) dễ đọc và dễ viết hơn.
* Tính năng:
  + Hỗ trợ viết các assertions theo cú pháp mượt mà và dễ hiểu, ví dụ như assertThat().
  + Đưa ra các thông báo lỗi chi tiết, giúp dễ dàng phát hiện lỗi khi test không thành công.
  + Hỗ trợ nhiều kiểu dữ liệu (đối tượng, tập hợp, chuỗi, số, v.v.).

### 4. TestNG

* Mô tả: TestNG là một framework kiểm thử mạnh mẽ lấy cảm hứng từ JUnit và NUnit, được phát triển để hỗ trợ kiểm thử đơn vị, kiểm thử tích hợp, và kiểm thử chức năng.
* Tính năng:
  + Hỗ trợ chạy song song các test case (parallel execution).
  + Hỗ trợ phụ thuộc giữa các test case, cho phép một số test chỉ chạy khi các test khác đã thành công.
  + Cung cấp khả năng nhóm các test (test groups) để tổ chức và phân loại các test case.
  + Hỗ trợ các chú thích (annotations) tương tự JUnit, nhưng cung cấp nhiều tính năng nâng cao như @DataProvider, @BeforeSuite, @AfterSuite.

### 5. Hamcrest

* Mô tả: Hamcrest là một thư viện cung cấp các matcher (trình so khớp), thường được sử dụng để mở rộng các assertion trong JUnit. Nó cung cấp một cú pháp mượt mà để kiểm tra tính đúng đắn của các đối tượng.
* Tính năng:
  + Cung cấp các matcher mạnh mẽ như is(), equalTo(), hasItems(), contains(), v.v.
  + Có thể kết hợp với JUnit hoặc các framework khác để làm cho việc viết kiểm thử dễ đọc và dễ hiểu hơn.

### 6. Powermock

* Mô tả: Powermock là một framework mở rộng Mockito và EasyMock, cho phép mock các thành phần khó khăn như các lớp tĩnh, các phương thức tĩnh, các phương thức private, và các khối khởi tạo tĩnh.
* Tính năng:
  + Hỗ trợ kiểm thử với các phương thức và đối tượng tĩnh.
  + Có thể sử dụng với Mockito để mở rộng khả năng mock.
  + Hỗ trợ mock các phương thức private và constructor.