

BÁO CÁO LAB 3 – KIỂM THỬ NÂNG CAO (JUnit 5)

Môn học: Kiểm thử nâng cao

Sinh viên: Bùi Tiến Sỹ

MSSV: PS32117

I. MỤC TIÊU LAB 3

- Hiểu và áp dụng vòng đồi kiểm thử trong JUnit 5.
- Thực hiện kiểm thử ngoại lệ trong chương trình Java.
- Áp dụng Parameterized Test để kiểm thử với nhiều bộ dữ liệu.

II. BÀI 1 – VÒNG ĐỒI KIỂM THỬ JUNIT

Trong bài 1, em đã sử dụng các annotation của JUnit 5 để quản lý vòng đồi kiểm thử bao gồm @BeforeAll, @BeforeEach, @AfterEach và @AfterAll.

Các annotation được sử dụng:

- @BeforeAll: Chạy một lần trước tất cả các test.
- @BeforeEach: Chạy trước mỗi test case.
- @AfterEach: Chạy sau mỗi test case.
- @AfterAll: Chạy một lần sau tất cả các test.

III. BÀI 2 – KIỂM THỬ NGOẠI LỆ

Trong bài 2, em sử dụng phương thức assertThrows() của JUnit 5 để kiểm thử các ngoại lệ phổ biến có thể xảy ra trong chương trình.

Các ngoại lệ đã kiểm thử:

- ArithmeticException: Xảy ra khi chia cho 0.
- NumberFormatException: Xảy ra khi chuyển đổi chuỗi không hợp lệ sang số.
- NullPointerException: Xảy ra khi truy cập đối tượng null.

Bài 1 và 2 gộp chung

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** Shows a project named "Lab3_PS32117_BuTienSy" with a "src" directory containing "main" and "test" packages. "main" contains "Calculator.java" and "NumberUtils.java". "test" contains "AppTest.java" and "NumberUtilsTest.java".
- Code Editor:** Displays the content of `AppTest.java`. It includes imports for `Calculator` and `NumberUtils`, and defines a static field `calculator`. It contains annotations `@BeforeAll` and `@BeforeEach` for setup logic, and a `@Test` method `testAdd()` that asserts the result of `calculator.add(2, 3)` is equal to 5.
- Run/Debug Tool Window:** Shows the results of a run named "AppTest (com.poly)". It indicates 5 tests passed in 18ms. The log output shows the execution of `Calculator` methods and the execution of `AppTest` methods (BeforeEach, AfterEach).
- Status Bar:** Shows the path "Lab3_PS32117_BuTienSy > src > test > java > com > poly > AppTest", along with file encoding (CRLF), character set (UTF-8), and code style (4 spaces).

IV. BÀI 3 – PARAMETERIZED TEST

Trong bài 3, em đã sử dụng Parameterized Test của JUnit 5 để kiểm thử chức năng kiểm tra số nguyên dương với nhiều bộ dữ liệu khác nhau.

Việc sử dụng `@ParameterizedTest` kết hợp với `@ValueSource` giúp giảm việc viết code lặp và tăng hiệu quả kiểm thử.

The screenshot shows the IntelliJ IDEA interface with a Java project named "Lab3_PS32117_BuTienSy". The project structure is visible on the left, showing packages like com.poly, com.poly.main, and com.poly.test, along with Java files such as Calculator.java, AppTest.java, NumberUtils.java, and NumberUtilsTest.java. The NumberUtilsTest.java file is currently open in the editor, displaying JUnit test code. On the right, the "Run" tool window is open, showing the results of a recent run. It indicates 7 tests passed (2 tests total, 31ms) and lists individual test cases: testNonPositiveNumbers(int) and testPositiveNumbers(int), each with three data points: [1] 5, [2] 10, and [3] 100. The status bar at the bottom shows the path as Lab3_PS32117_BuTienSy > src > test > java > com > poly > NumberUtilsTest > testPositiveNumbers.

```
package com.poly;

import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;
import static org.junit.jupiter.api.Assertions.*;

public class NumberUtilsTest {
    // Chia nhỏ 1 test thành nhiều lần với dữ liệu khác nhau
    @ParameterizedTest
    // Sử dụng danh sách giá trị test
    @ValueSource(ints = {1, 5, 10, 100})
    void testPositiveNumbers(int number) {
        assertTrue(NumberUtils.isPositive(number));
    }

    @ParameterizedTest
    @ValueSource(ints = {0, -1, -10})
    void testNonPositiveNumbers(int number) {
        assertFalse(NumberUtils.isPositive(number));
    }
}
```

V. KẾT LUẬN

Thông qua Lab 3, em đã nắm được cách sử dụng JUnit 5 để thực hiện kiểm thử nâng cao, bao gồm vòng đời kiểm thử, kiểm thử ngoại lệ và kiểm thử có tham số. Lab giúp em hiểu rõ hơn về tầm quan trọng của kiểm thử trong phát triển phần mềm.