BÁO CÁO LAB 2

SOF304 – Kiểm thử nâng cao



Nhóm 2:

Phạm Minh Đức

Huỳnh Tấn Phát

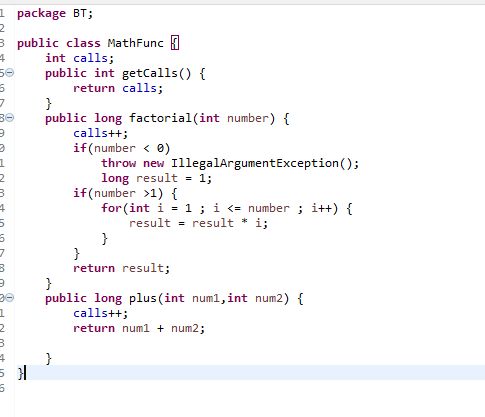
Lê Ngọc Khánh

Lưu Hoàng Bảo

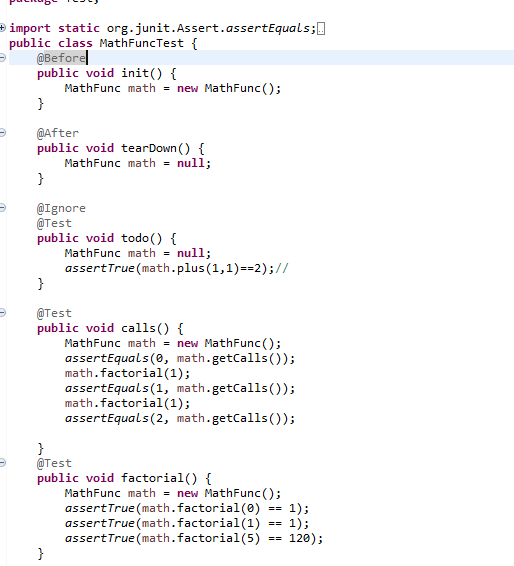
Trịnh Phùng Hồng Phúc

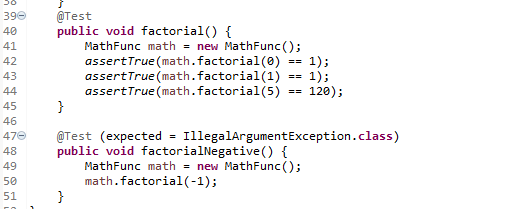
Bài 1 (2 điểm) Sinh viên thực hiện xây dựng 1 chương trình kiểm thử ứng dụng có 2 phương thức là tính giai thừa một số nguyên không âm và tính tổng 2 số nguyên, có sử dụng test fixture

* Tạo class có tên MathFunc, xây dưng 2 phương thức factorial(tính giai thừa 1 số nguyên) và plus(tính tổng 2 số tự nhiên):

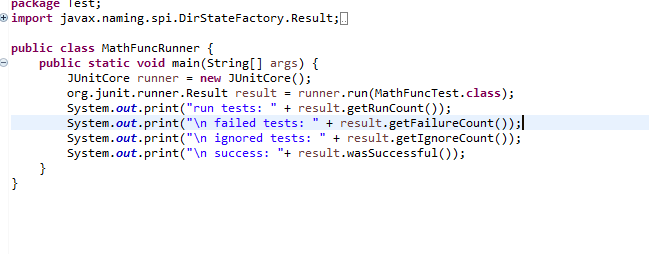


* Tạo junit test case, tạo class có tên MathFuncTest chứa các test case cần thiết để test 2 phương thức bên trên.

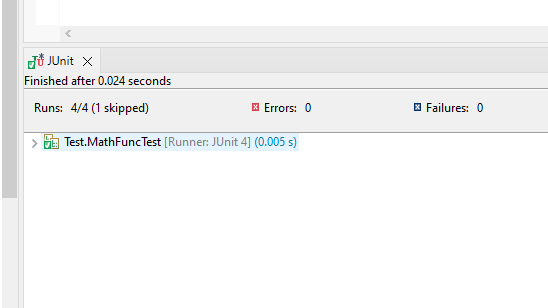




* Phương thức public void factorial() kiểm tra phương thức tính giai thừa
* 0! = 1 => Đúng
* 1! = 1 => Đúng
* 5! = 120 => Đúng
* Phương thức factorialNegative() kiểm tra trường hợp số âm với expected(mong muốn) trả về ngoại lệ
* Truyền vào -1 => Ngoại lệ
* Phương thức public void todo() kiểm tra phướng thức tính tổng, nhưng với @Ignore thì bỏ qua Testcase này
* Tạo class thực thi có chưa main như sau:



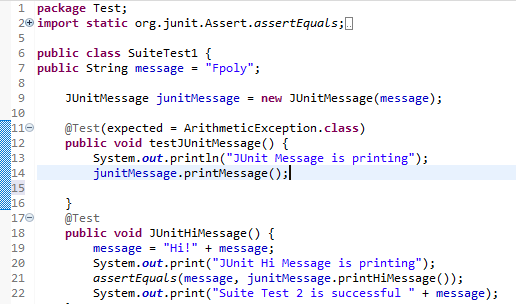
* Chạy kiểm tra kết quả:



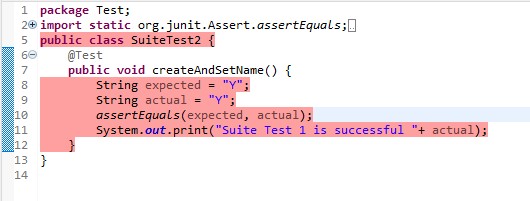
* Số test kiểm tra: 3
* Số test failed: 0
* Số test bỏ qua: 1
* Thành công: Đúng

Bài 2 (3 điểm) Tạo JUnit Test Suite sử dụng @RunWith @Suite. Giả sử có 2 lớp SuiteTest1.java và SuiteTest2.java, mỗi class chứa những test case khác nhau.

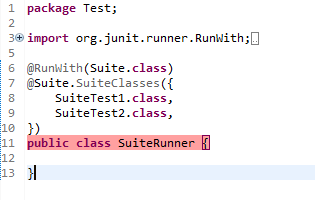
* Tạo class SuiteTest1.java với mục đích xuất ra một chuỗi ký tự



* Tương tự tạo class SuiteTest2.java



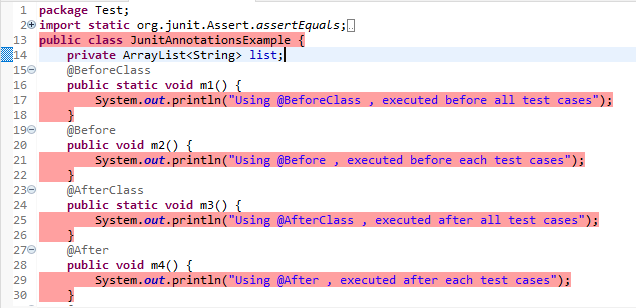
* Tạo class JunitTest.java để thực thi 2 lớp Suitest bên trên

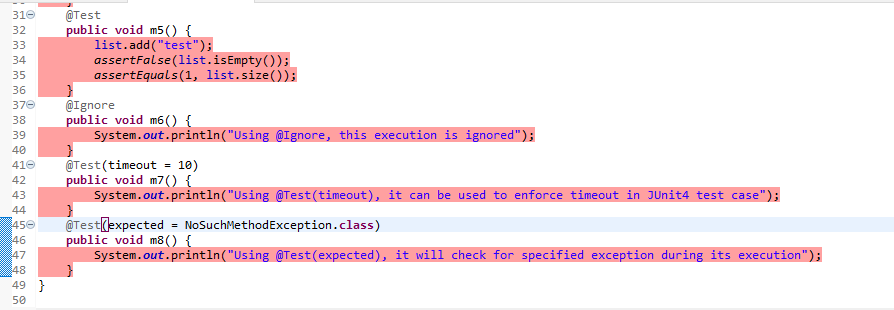


* @RunWith(Suite.class) cho phép gom chạy cùng lúc nhiều test case có trình tự
* Chạy test và kiểm tra kêt quả

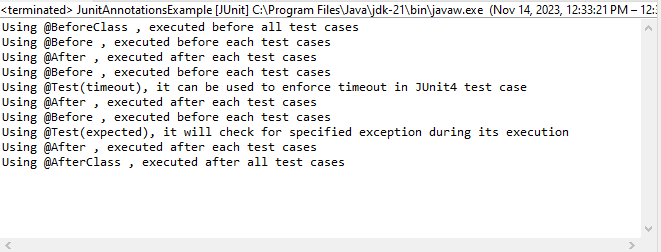
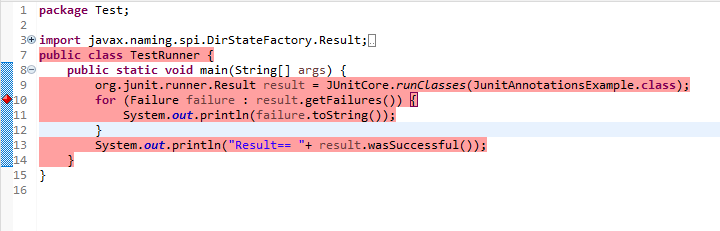
Bài 3 (3 điểm) Xây dựng lớp kiểm thử với JUnit Annotations: @Before , @BeforeClass, @After, @AfterClass, @Test, @Ignore…

* Tạo lớp kiểm thử JunitAnnotationsExample.java





* Tạo lớp TestRunner.java thực thi đoạn test case bên trên
* Chạy và kiểm tra kết quả



* Sử dụng @BeforeClass để thực thi trước tất cả TestCase
* Sử dụng @Before để thực thi trước mỗi TestCase
* Sử dụng @AfterClass để thực thi sau tất cả TestCase
* Sử dụng @After để thực thi sau mỗi TestCase
* Sử dụng @Test(timeout = ?) để đặt thời gian timeout cho TestCase
* Sử dụng @Test(expected=?) để check trường hợp ngoại lệ mong muốn