



KIỂM THỬ NÂNG CAO

BÀI 8: TEST NEXT GENERATION

- ① Annotations
- ① Data-driven testing



Phần I: Annotations

-  Các loại annotation

-  Cấu hình annotation

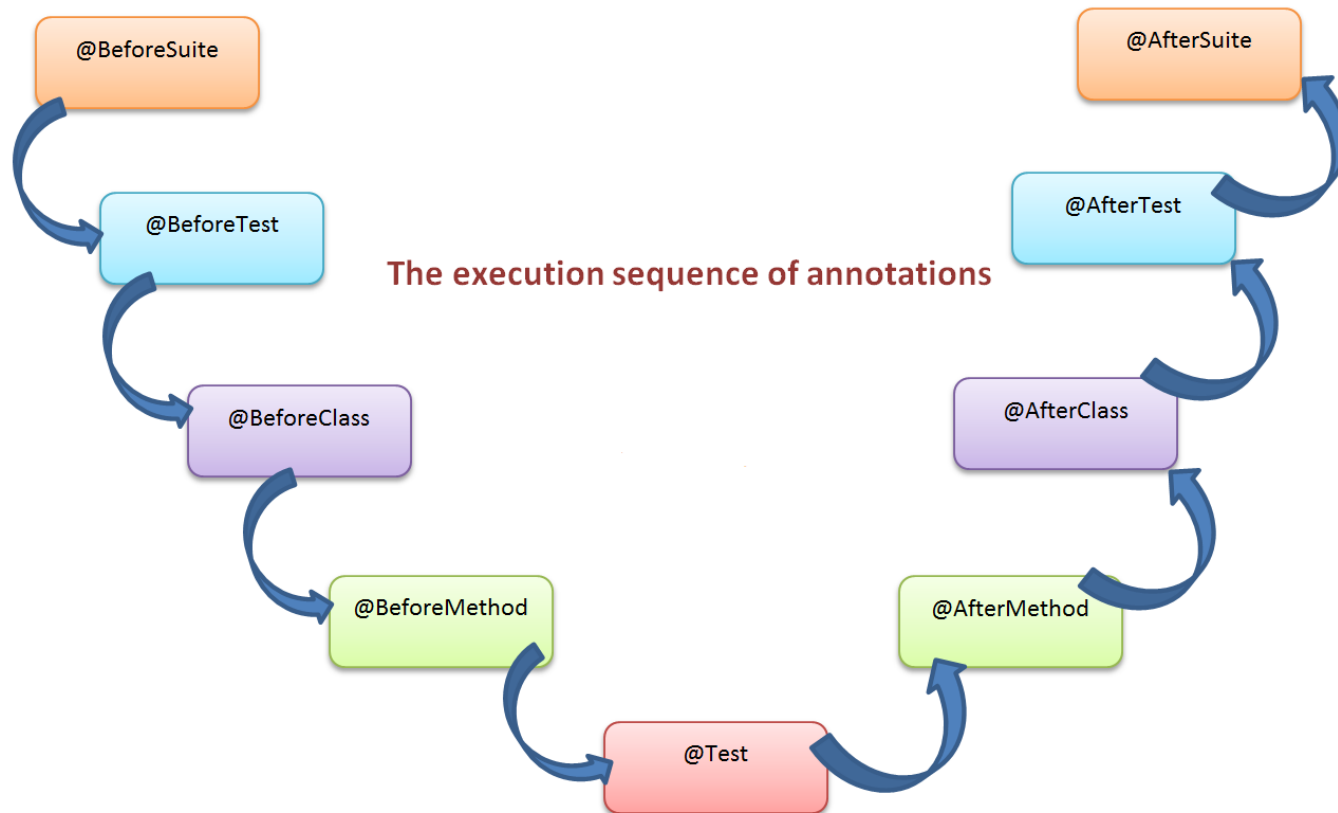
Phần II: Data-driven testing

-  Khái niệm data-driven

-  Ứng dụng annotation trong data-driven



- ❑ Là các chỉ thị khai báo chung với lớp, phương thức, biến trong code java
- ❑ Giúp làm chủ quá trình thực thi test case tốt hơn



- ❑ @BeforeTest : Test case sử dụng annotation này sẽ thực thi trước hết trong tất cả các test case
- ❑ @AfterTest: Test case sử dụng annotation này sẽ thực thi sau cùng trong tất cả các test case
- ❑ Xét ví dụ có 3 test case: launchBrowser, verifyHomepageTitle và terminateBrowser

```
package fpoly;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.*;
public class firsttestngfile {
    public String baseUrl = "http://http://lms.poly.edu.vn/";
    String driverPath = "C:\\\\geckodriver.exe";
    public WebDriver driver ;

    @BeforeTest
    public void launchBrowser() {
        System.out.println("launching firefox browser");
        System.setProperty("webdriver.firefox.marionette", driverPath);
        driver = new FirefoxDriver();
        driver.get(baseUrl);
    }
    @Test
    public void verifyHomepageTitle() {
        String expectedTitle = "Hệ thống quản lý trị học tập Fpt PolyTech
nic";

        String actualTitle = driver.getTitle();
        Assert.assertEquals(actualTitle, expectedTitle);
    }
    @AfterTest
    public void terminateBrowser(){
        driver.close();
    }
}
```

- ❑ Thứ tự thực thi các test case là : launchBrowser() -> verifyHomepageTitle() -> terminateBrowser()

Methods in chronological order

firsttestngpackage.FirstTestNGFile

launchBrowser	0 ms
verifyHomepageTitle	4641 ms
terminateBrowser	4664 ms

- ❑ @BeforeMethod: test case sử dụng annotation này sẽ thực thi trước mỗi một test case có trong lớp test
- ❑ @AfterMethod: test case sử dụng annotation này sẽ thực thi ngay sau mỗi một test case có trong lớp test
- ❑ Cho biết thứ tự thực thi lớp kiểm thử có các test case và annotation như sau?

```
@BeforeTest
public void launchBrowser() {

    @BeforeMethod
    public void verifyHomepageTitle() {

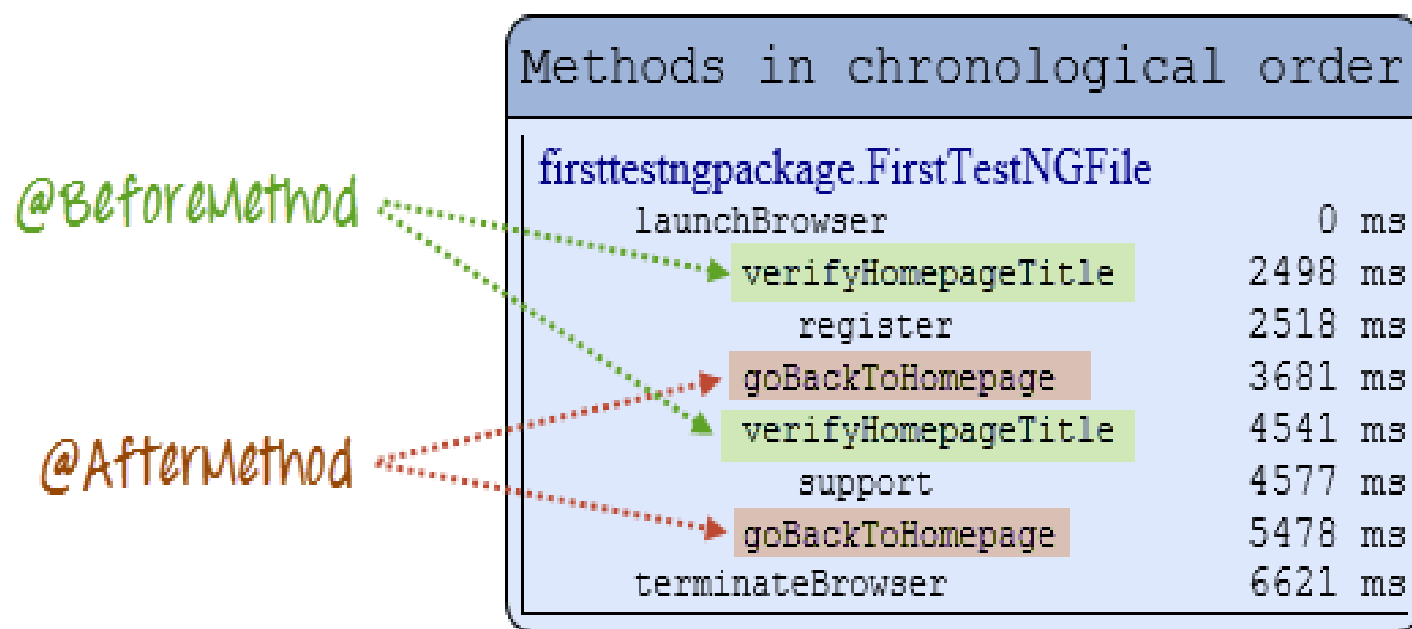
        @Test(priority = 0)
        public void register(){

            @Test(priority = 1)
            public void support() {

                @AfterMethod
                public void goBackToHomepage ( ) {

                    @AfterTest
                    public void terminateBrowser(){
```


❑ Kết quả thực thi lớp kiểm thử trên:





DEMO

- Demo @BeforeMethod,
@AfterMethod

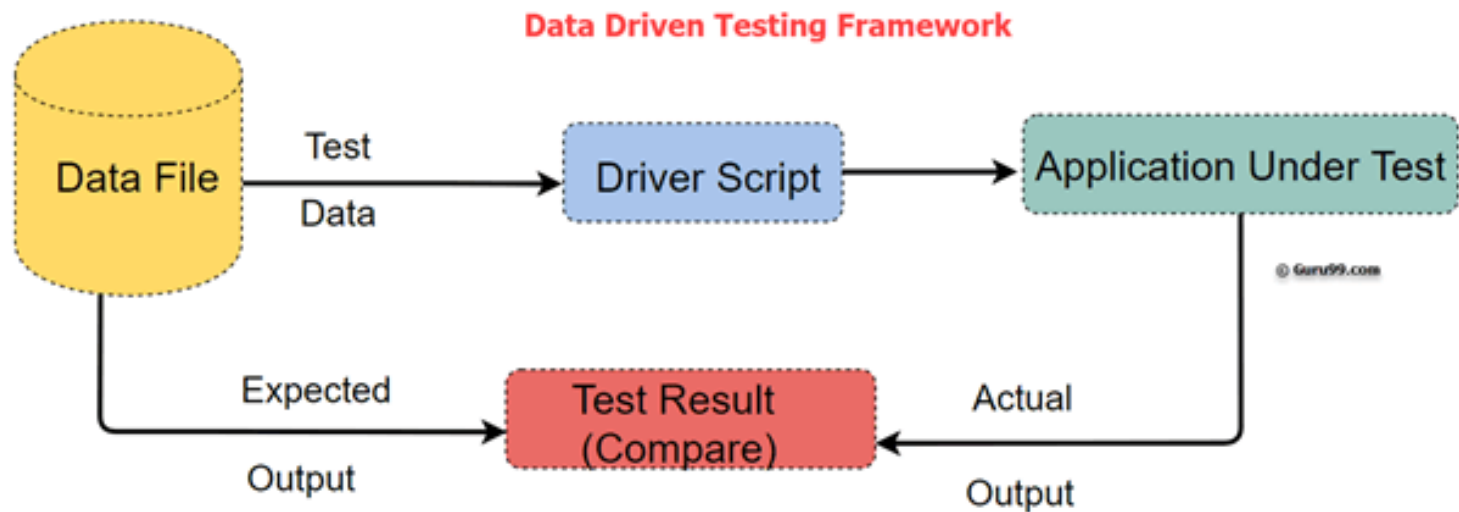




KIỂM THỬ NÂNG CAO

BÀI 8: TEST NEXT GENERATION (P2)

- ❑ Một test case có thể thực thi nhiều lần với data khác nhau->tách rời test data với test script -> data-driven
- ❑ TestNG hỗ trợ 2 cơ chế là TestNG Parameters và DataProvider annotations
- ❑ Test data lưu bên ngoài có thể là tập tin, excel...



□ Annotation <@Parameters>

- ❖ Truyền đối số tại thời điểm thực thi
- ❖ Phù hợp khối lượng dữ liệu nhỏ, ít thay đổi
- ❖ Có thể truyền một hoặc nhiều đối số
- ❖ Ví dụ tạo lớp kiểm thử có 2 test case:

➤ Test case nhận một @Parameters

```
@Parameters({ "webPoly" })  
@Test  
public void OpenBrowser(String webPoly) {
```

➤ Test case nhận hai @Parameters

```
@Parameters({ "UserName", "Passcode" })  
@Test  
public void FillLoginForm(String UserName, String Passcode)
```

Annotation <@Parameters>

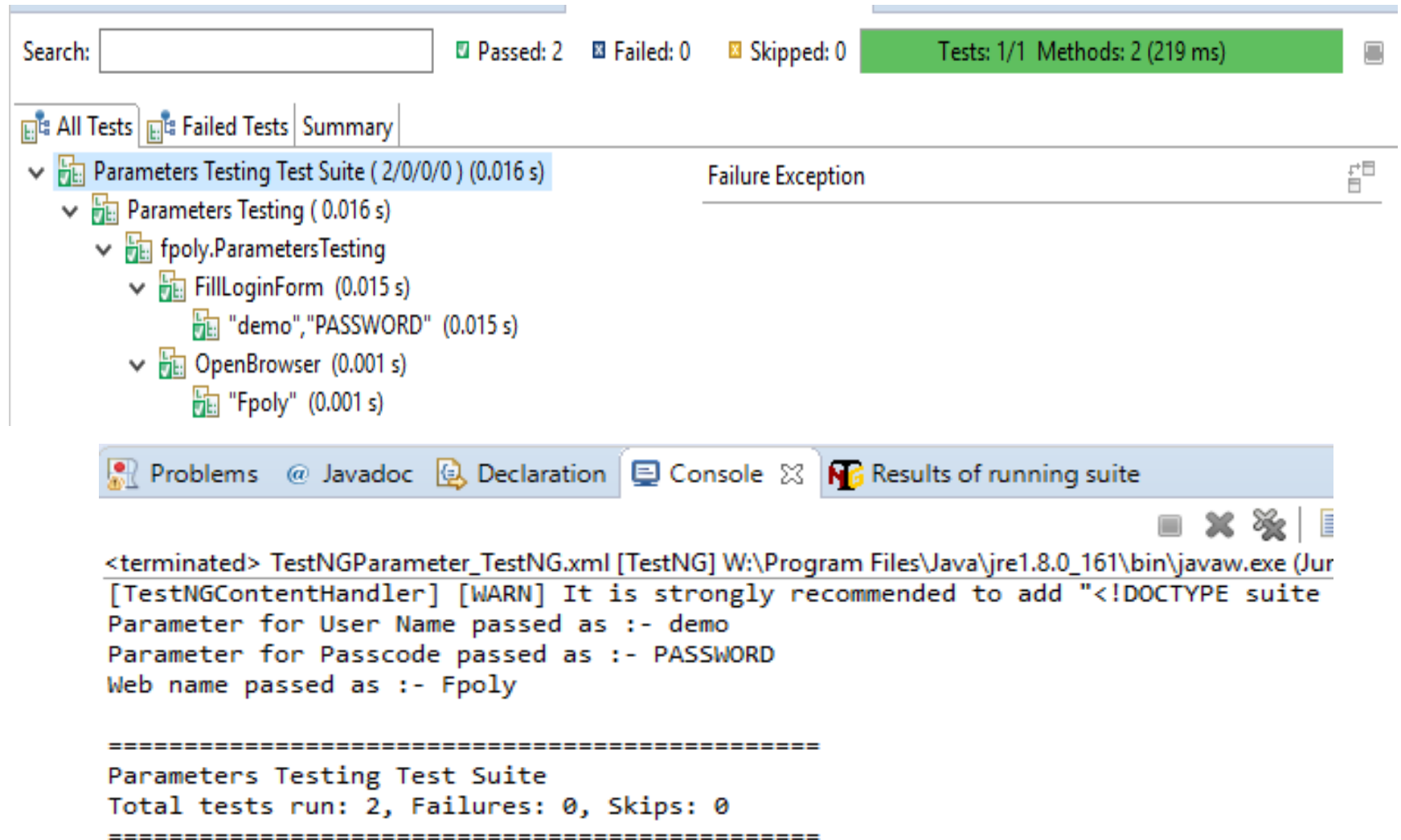
```
1 package fpoly;
2 import org.testng.annotations.Parameters;
3 import org.testng.annotations.Test;
4 public class ParametersTesting {
5     @Parameters({ "webPoly" })
6     @Test
7     public void OpenBrowser(String webPoly) {
8         System.out.println("Web name passed as :- " + webPoly);
9     }
10
11     @Parameters({ "UserName", "Passcode" })
12     @Test
13     public void FillLoginForm(String UserName, String Passcode) {
14         System.out.println("Parameter for User Name passed as :- " + UserName);
15         System.out.println("Parameter for Passcode passed as :- " + Passcode);
16     }
17 }
18
```

□ Annotation <@Parameters>

- ❖ Tạo tập tin TestNG.xml cấu hình truyền giá trị cho các tham số

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <suite name="Parameters Testing Test Suite">
3     <test name="Parameters Testing">
4         <parameter name="webPoly" value="Fpoly"/>
5         <parameter name="UserName" value="demo"/>
6         <parameter name="Passcode" value="PASSWORD"/>
7         <classes>
8             <class name="fpoly.ParametersTesting" />
9         </classes>
10    </test>
11 </suite>
```

Annotation <@Parameters>



The screenshot displays the TestNG results in an IDE. At the top, a search bar is followed by summary statistics: **Passed: 2**, **Failed: 0**, and **Skipped: 0**. A green bar on the right indicates **Tests: 1/1 Methods: 2 (219 ms)**. Below this, tabs for **All Tests**, **Failed Tests**, and **Summary** are visible. The test hierarchy is expanded, showing the following structure:

- Parameters Testing Test Suite (2/0/0/0) (0.016 s)
 - Parameters Testing (0.016 s)
 - fpoly.ParametersTesting
 - FillLoginForm (0.015 s)
 - "demo", "PASSWORD" (0.015 s)
 - OpenBrowser (0.001 s)
 - "Fpoly" (0.001 s)

The **Failure Exception** column is empty, indicating all tests passed. At the bottom, the **Results of running suite** tab is active, showing the following output:

```
<terminated> TestNGParameter_TestNG.xml [TestNG] W:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (J  
[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite  
Parameter for User Name passed as :- demo  
Parameter for Passcode passed as :- PASSWORD  
Web name passed as :- Fpoly  
  
=====  
Parameters Testing Test Suite  
Total tests run: 2, Failures: 0, Skips: 0  
=====
```


❑ <@Optional> annotation

- ❖ Là một dạng optional parameter
- ❖ Có thể nhận giá trị mặc định nếu lúc thực thi không truyền giá trị
- ❖ Cấu hình bên trong file xml
- ❖ Ví dụ: Xây dựng lớp kiểm thử tên OptionalTesting gồm test case tên optionCheck

```
package com.fpoly.testng;  
  
import org.testng.annotations.Optional;  
import org.testng.annotations.Parameters;  
import org.testng.annotations.Test;  
  
public class OptionalTesting  
{  
    @Parameters({ "optional-value" })  
    @Test  
    public void optionCheck(@Optional("Optional Value") String value) {  
        System.out.println("This is: " + value);  
    }  
}
```

❑ <@Optional> annotation

- ❖ Tạo file TestNG.xml, thiết lập cho test case chạy 2 lần

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://fpoly.edu.vn/testng-1.0.dtd">
<suite name="Optional Testing Test Suite" verbose="1">
  <test name="Optional Testing Part1">
    <classes>
      <class name="com.fpoly.testng.OptionalTesting" />
    </classes>
  </test>
  <test name="Optional Testing Part2">
    <parameter name="optional-value" value="Provided from XML" />
    <classes>
      <class name="com.fpoly.testng.OptionalTesting" />
    </classes>
  </test>
</suite>
```

❑ <@Optional> annotation

- ❖ Thực thi lớp test cho kết quả lần thứ 1 test case nhận giá trị mặc định
- ❖ Lần 2 nhận giá trị thiết lập từ file xml

```
[TestNG] Running:
  C:\TestNGSeleniumDemo\optional\testng.xml

This is: Optional Value
This is: Provided from XML

=====
Optional Testing Test Suite
Total tests run: 2, Failures: 0, Skips: 0
=====
```



DEMO

- Demo @Parameters



□ Annotation <priority>

- ❖ Xác định thứ tự thực thi của các test case
- ❖ Mức độ ưu tiên thấp sẽ ưu tiên thực hiện trước

Priorities

①

②

③



```
@Test(priority = 0)
```

parameter

value of the
parameter

□ Annotation <priority>

- ❖ Ví dụ tạo lớp kiểm thử theo trình tự: đăng ký thành công->đăng nhập- gửi mail

```
import org.testng.annotations.Test;

public class testNGPriorityExample {

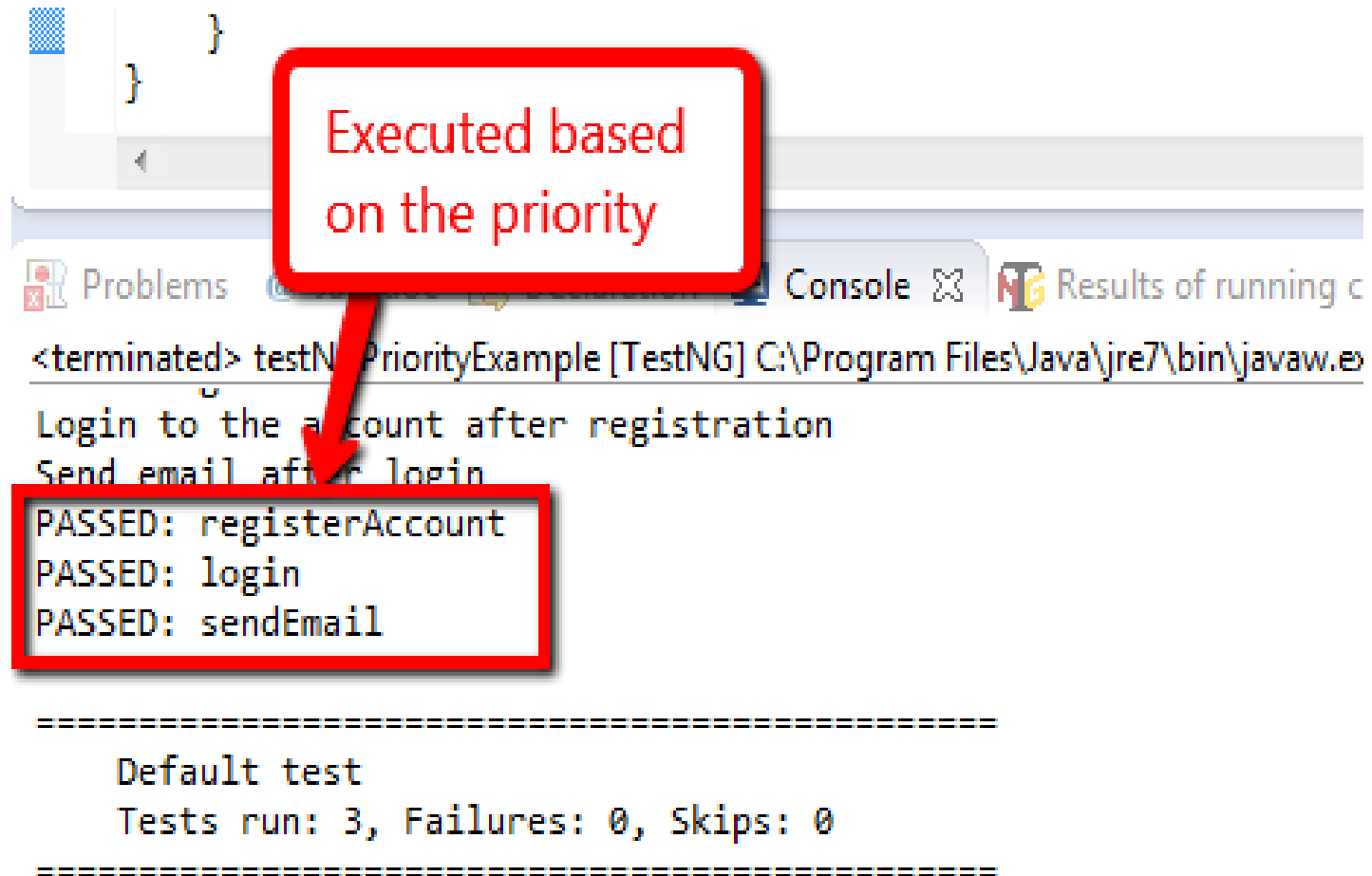
    @Test
    public void registerAccount()
    {
        System.out.println("First register your account");
    }

    @Test(priority=2)
    public void sendEmail()
    {
        System.out.println("Send email after login");
    }

    @Test(priority=1)
    public void login()
    {
        System.out.println("Login to the account after registration");
    }

}
```


Annotation <priority>




```
}  
}  
  
Problems Console Results of running c  
<terminated> testN PriorityExample [TestNG] C:\Program Files\Java\jre7\bin\javaw.e  
Login to the account after registration  
Send email after login  
PASSED: registerAccount  
PASSED: login  
PASSED: sendEmail  
  
=====  
Default test  
Tests run: 3, Failures: 0, Skips: 0  
=====
```

Tổng kết bài học

Phần I: Annotations

 Các loại annotation

 Cấu hình annotation

Phần II: Data-driven testing

 Khái niệm data-driven

 Ứng dụng annotation trong data-driven





KẾT THÚC