

بسم الله الرحمن الرحيم

آزمایش سوم در آز مهندسی نرم افزار

سید مهدی فقیه 97106198

محمدرضا یوسف پور 97106324

میزان مشارکت هر دو عضو تیم به صورت یکسان بوده است.

ادرس گیت هاب : <https://github.com/SMahdiFaghih/SoftwareEngineeringLab-E3>

تست های undefined :

در اجرای مجدد RunnerTest یکی از تست ها به مشکل undefined برخورد کرد که تست دوم بود که یکی از عملوند های آن عددی منفی بود. اعداد تست | 5 | 6 | -1 | بودند که به دلیل منفی بودن یکی از عملوند ها به مشکل خورده است.

علت این مشکل همانطور که گفته شد برای منفی بودن این عدد است که در فایل MyStepdefs در تگ Given

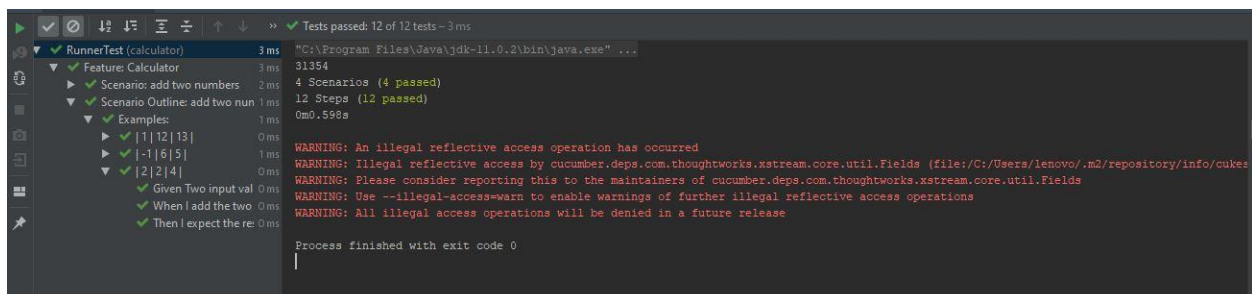
```
@Given("^Two input values, (\\d+) and (\\d+)$")
```

ریجکس `\\d+` بیانگر اعداد نا منفی هستند یعنی تنها اعداد نا منفی را می تواند قبول کند برای حل این مشکل باید به اول ریجکس این را اضافه کنیم که کاراکتر '-' را نیز بتواند قبول کند بنابراین ریجکس نهایی `[-]?\\d+` است که در ابتدای خود می تواند 0 یا 1 کاراکتر منفی قبول کند.

حال تغییرات به صورت زیر است

```
@Given("^Two input values, ([-]?\\d+) and ([-]?\\d+)$")
```

که نتیجه ی زیر را دارد.



```
Tests passed: 12 of 12 tests - 3 ms
RunnerTest (calculator) 3 ms
  Feature: Calculator 3 ms
    Scenario: add two numbers 2 ms
    Scenario Outline: add two numbers 1 ms
      Examples: 1 ms
        1 | 12 | 13 | 0 ms
        2 | 1 | 6 | 5 | 1 ms
        3 | 2 | 2 | 4 | 0 ms
        Given Two input values 0 ms
        When I add the two 0 ms
        Then I expect the result 0 ms
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by cucumber.deps.com.thoughtworks.xstream.core.util.Fields (file:/C:/Users/lenovo/.m2/repository/info/cukes...)
WARNING: Please consider reporting this to the maintainers of cucumber.deps.com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Process finished with exit code 0
```

برای انجام سناریوی عادی مسئله ی آزمایش ما ابتدا در فایل calculator.feature سناریوی زیر را اضافه کردیم.

```
@tag
Feature: Calculator
# Scenario: add two numbers
#   Given Two input values, 1 and 2
#   When I add the two values
#   Then I expect the result 3
#
# Scenario Outline: add two numbers
#   Given Two input values, <first> and <second>
#   When I add the two values
#   Then I expect the result <result>
#
# Examples:
#   | first | second | result |
#   | 1 | 12 | 13 |
#   | -1 | 6 | 5 |
#   | 2 | 2 | 4 |
Scenario: squareRoot of division of a by b
  Given Two input values, 4 and 1
  When I get squareRoot of division of values
  Then I expect the result 2
```

که با Alt + Enter زدن بر روی هر کدام از خطوط یک فایلی به نام MyStepdefs2 ایجاد می شود که به صورت تصویر زیر است.

```

package calculator;

import cucumber.api.java.Before;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;
import org.junit.Assert;

public class MyStepdefs2 {

    private Calculator calculator;
    private int value1;
    private int value2;
    private int result;

    @Before
    public void before() {
        calculator = new Calculator();
    }

    @Given("^Two input values, (\\d+) and (\\d+)$")
    public void twoInputValuesAnd(int arg0, int arg1) {
        value1 = arg0;
        value2 = arg1;
    }

    @When("^I get squareRoot of division of values$")
    public void iGetSquareRootOfDivisionOfValues() {
        result = calculator.sqrtDivide(value1, value2);
        System.out.print(result);
    }

    @Then("^I expect the result (\\d+)$")
    public void iExpectTheResult(int arg0) {
        Assert.assertEquals(arg0, result);
    }

}

```

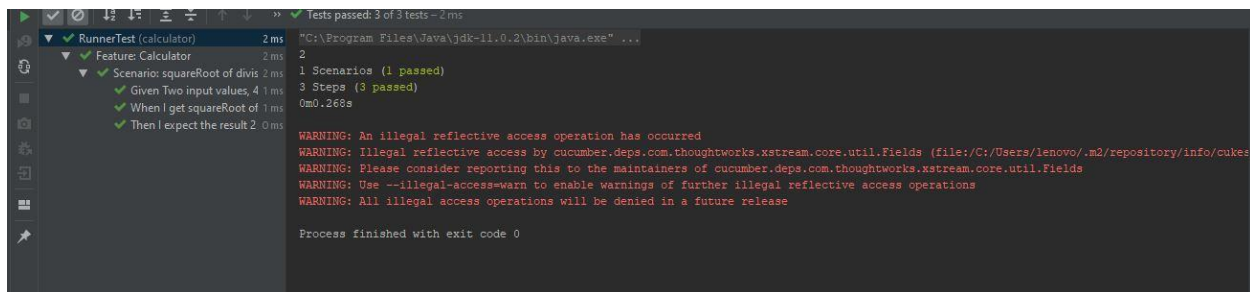
حال در تابع `iGetSquareRootOfDivisionOfValues` مشاهده می کنید که تابع `sqrtDivide` داریم که به صورت زیر است (در فایل `Calculator.java`)

```

public int sqrtDivide(int a, int b){
    return (int)Math.sqrt(a/b);
}

```

حال در `RunnerTest` اجرا کردیم که نتیجه ی موفقیت آمیز زیر را داشت.



حال سناریوی outline را اضافه می کنیم.

```
Scenario Outline: squareRoot of division of a by b
  Given Two input values, <first> and <second>
  When I get squareRoot of division of values
  Then I expect the result <result>

Examples:
  | first | second | result |
  | 4     | 1      | 2      |
  | 36    | 4      | 3      |
```

سپس مجددا RunnerTest را اجرا می کنیم که نتیجه ی موفقیت آمیز زیر را دارد.

