

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu được các khái niệm liên quan ngoại lệ
- ✓ Kỹ thuật xử lý, quản lý error trong junit
- ✓ Kỹ thuật dùng Parameters

Bài 1 (2 điểm)

Tạo class kiểm thử có sử dụng ngoại lệ với @Test(expected).

- Tạo class tên JUnitMessage chứa xử lý phép chia cho 0

```
1 package fploy;
2 public class JUnitMessage {
3     private String message;
4
5     public JUnitMessage(String message) {
6         this.message = message;
7     }
8
9     public void printMessage(){
10
11         System.out.println(message);
12         int divide=1/0;
13
14     }
15
16     public String printHiMessage(){
17         message="Hi!" + message;
18
19         System.out.println(message);
20
21         return message;
22     }
23 }
24
25
26
```

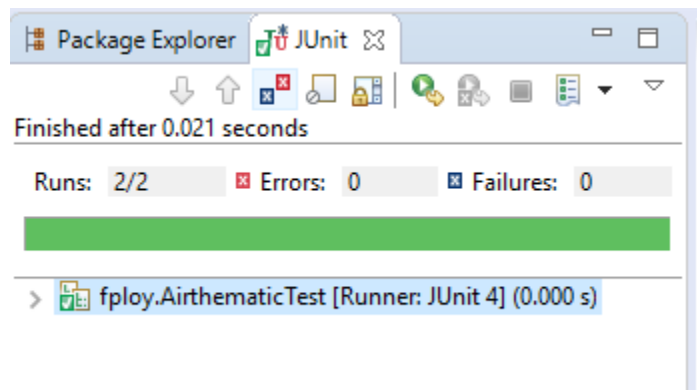
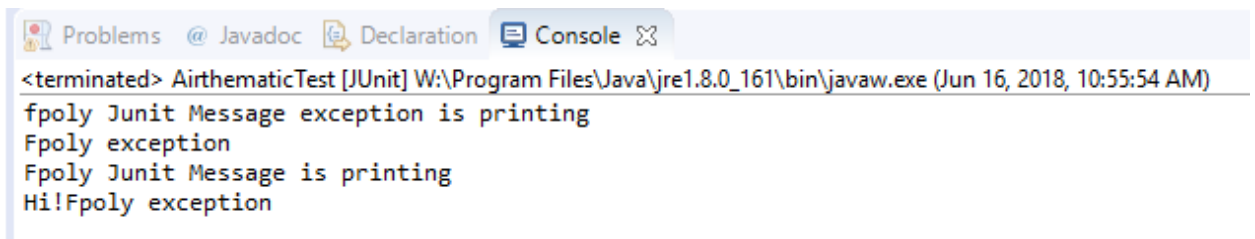
- Dễ dàng nhận thấy phép chia cho 0 sẽ gây ra ngoại lệ, ta thiết kế lớp kiểm thử quản lý và bắt ngoại lệ này.
- Tạo class tên AirthematicTest có dùng @Test annotation

```

1 package fpjoy;
2 import static org.junit.Assert.assertEquals;
3
4 import org.junit.Test;
5 public class AirthematicTest {
6
7     public String message = "Fpoly exception";
8
9     JUnitMessage junitMessage = new JUnitMessage(message);
10
11     @Test(expected = ArithmeticException.class)
12     public void testJUnitMessage() throws Exception{
13
14         System.out.println("fpjoy Junit Message exception is printing ");
15         junitMessage.printMessage();
16     }
17
18     @Test
19     public void testJUnitHiMessage(){
20         message="Hi!" + message;
21         System.out.println("Fpoly Junit Message is printing ");
22         assertEquals(message, junitMessage.printHiMessage());
23     }
24 }
25
26

```

- Chạy lớp test thì kết quả ngoại lệ đã được đón bắt thành công, cho test pass

The screenshot shows the console output of the test. It displays the following text:

<terminated> AirthematicTest [JUnit] W:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Jun 16, 2018, 10:55:54 AM)

fpjoy Junit Message exception is printing

Fpoly exception

Fpoly Junit Message is printing

Hi!Fpoly exception

- f

Bài 2 (3 điểm)

Tạo class kiểm thử có sử dụng ngoại lệ với @Test(expected), ExpectedException Rule, try-catch

- Tạo class tên Person, có 2 field là Name và age

```
package fpoloy;
public class Person{
    private final String name;
    private final int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
        if (age <= 0) {
            throw new IllegalArgumentException("Invalid age:" + age);
        }
    }
}
```

- Class Person sử dụng constructor để gọi ngoại lệ IllegalArgumentException khi age<0.
- Sử dụng “ExpectedException Rule” trong lớp test để theo dõi ngoại lệ

```

import static org.junit.Assert.*;

import org.junit.Rule;

import org.junit.Test;

import org.junit.rules.ExpectedException;

public class PersonTest {
    @Rule

    public ExpectedException exception = ExpectedException.none();

    @Test

    public void testExpectedException() {

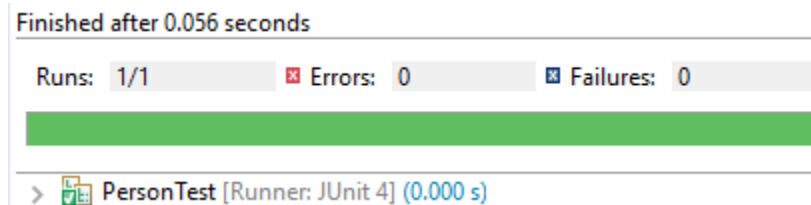
        exception.expect(IllegalArgumentException.class);

        new Person("Fpoly", -1);

    }
}

```

Kết quả chạy test case trên



- Sử dụng “@Test annotation” trong lớp test để theo dõi ngoại lệ



```


5 import static org.junit.Assert.*;
6
7 import org.junit.Rule;
8
9 import org.junit.Test;
10
11 import org.junit.rules.ExpectedException;
12
13 public class PersonTest {
14     @Test(expected = IllegalArgumentException.class)
15
16     public void testExpectedException2() {
17
18         new Person("Fpoly", -1);
19
20     }
21
22 }
23

```

Kết quả chạy test case

Finished after 0.027 seconds

Runs: 1/1  Errors: 0  Failures: 0

>  PersonTest [Runner: JUnit 4] (0.000 s)

- Sử dụng khối “try-catch” trong lớp test để theo dõi ngoại lệ

```

import static org.junit.Assert.*;

import org.hamcrest.Matcher;
import org.junit.Rule;

import org.junit.Test;

import org.junit.rules.ExpectedException;

public class PersonTest {
    @Test
    public void testExpectedException3() {

        try {

            new Person("Fpoly", -1);

            fail("Should have thrown an IllegalArgumentException because age is invalid!");

        } catch (IllegalArgumentException e) {



        }

    }
}

```

Kết quả chạy test case

Finished after 0.028 seconds

Runs: 1/1  Errors: 0  Failures: 0

>  PersonTest [Runner: JUnit 4] (0.000 s)

Bài 3 (3 điểm)

Xây dựng lớp kiểm thử với ErrorCollector @Rule

- Tạo lớp kiểm thử tên ErrorCollectorExample, sử dụng addError(throwable) thu thập lỗi.

```
package fpoly.junit;

import org.junit.Assert;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ErrorCollector;

public class ErrorCollectorExample {
    @Rule
    public ErrorCollector collector = new ErrorCollector();

    @Test
    public void example() {
        collector.addError(new Throwable("There is an error in first line"));

        collector.addError(new Throwable("There is an error in second line"));

        System.out.println("Hello");
        try {
            Assert.assertTrue("A " == "B");

        } catch (Throwable t) {
            collector.addError(t);
        }
        System.out.println("World!!!!");
    }
}
```

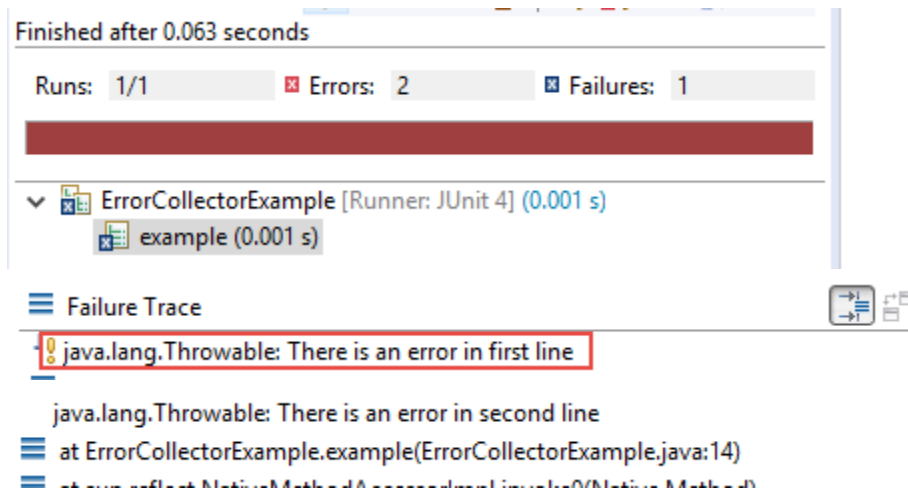
- Tạo lớp TestRunner.

```

1
2 import org.junit.runner.JUnitCore;
3 import org.junit.runner.Result;
4 import org.junit.runner.notification.Failure;
5
6 public class TestRunner {
7     public static void main(String[] args) {
8         Result result = JUnitCore.runClasses(ErrorCollectorExample.class);
9         for (Failure failure : result.getFailures()) {
10             System.out.println(failure.toString());
11         }
12         System.out.println("Result==" + result.wasSuccessful());
13     }
14 }

```

Chạy testcase và nhận thấy các lỗi được lưu vết



➤ sd

Bài 4 (2 điểm)

Giảng viên cho thêm