Junit 参数化测试

参数化测试就好比把一个“输入值，期望值”的集合传入给测试方法，达到一次性测试的目的。以第一次实验 EXP1Test 为例：

```
package com.softtest.baihe.test;

import com.softtest.baihe.EXP1;

import static org.junit.Assert.*;
import org.junit.Test;

//1、参数化测试：引入相关的包和类
import java.util.Collection;
import java.util.Arrays;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;

@RunWith(Parameterized.class) //2、参数化测试：更改测试运行器为 RunWith(Parameterized.class)
public class EXP1Test {
    //3、参数化测试：声明变量用来存放预期值与结果值
    private EXP1 exp1 = new EXP1();
    private int x=0,y=0,z=0;
    private int j = 0;

    //4、参数化测试：声明一个返回值为 Collection 的公共静态方法，并使用@Parameters 进行修饰
    @Parameters
    public static Collection data(){
            return Arrays.asList(new Object[][]{
            {4, 5, 8, 0},
            {4, 5, 3, 0},
            {2, 6, 8, 1},
            {2, 4, 11, 0}
            });
    }

    //5、参数化测试：为测试类声明一个带有参数的公共构造方法，并在其中为声明变量赋值
    public EXP1Test(int x,int y,int z,int j){
        this.x = x;
```

```
            this.y = y;
            this.z = z;
            this.j = j;
        }

        @Test
        public void testDoWork2() {
            assertEquals(j, exp1.doWork(x, y, z));
        }
}
```