**实验三 单元测试及JUnit的应用**

班级：信1405-1班 学号：20142863 姓名：杨程鑫

**一、实验目的**

1、掌握单元测试的基本理论和作用。

2、掌握典型单元测试工具 JUnit 的使用。

**二、实验类型**

验证加设计。

**三、实验内容**

请按照以下关于单元测试的讲解说明，完成两个测试任务（即“实验任务 1 ”和“实验任务 2”），并在有时间的情况下，阅读后面关于单元测试和”），并在有时间的情况下，阅读后面关于单元测试和JUnit 高级特性的内容，进一步了解单元测试的理论和JUnit测试的应用。

**实验任务1**

**源代码：**

package test3\_1;

public class File

{

public static void main(String[] args) throws Exception

{

FileTest Test = new FileTest ();

Test.setUp();

Test.testAbs();

Test.tearDown();

}

public int abs(int n)

{

if(n >= 0)

return n;

else

return (-n);

}

}

package test3\_1;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class FileTest

{

private File file;

@Before

public void setUp() throws Exception

{

file = new File();

}

@After

public void tearDown() throws Exception

{

}

@Test

public void testAbs()

{

assertEquals(file.abs(14),14);

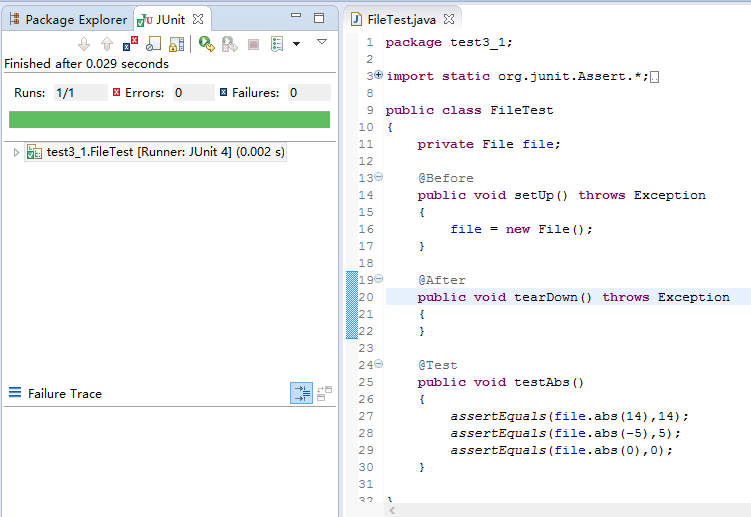
assertEquals(file.abs(-5),5);

assertEquals(file.abs(0),0);

}

}

**运行截图**

****

**实验任务2**

**源代码：**

package test3\_2;

public class Calculator

{

private static int result; // 静态变量，用于存储运行结果

public void add(int n)

{

result = result + n;

}

public void substract(int n)

{

result = result - n; // 故意的Bug，应该是 result =result-n

}

public void multiply(int n)

{

} // 假设此方法在项目完成过程中尚未写好

public void divide(int n)

{

result = result / n;

}

public void square(int n)

{

result = n \* n;

}

public void squareRoot(int n)

{ // 求平方根

for(;;)

; // Bug : 死循环

}

public void clear()

{ // 将结果清零

result = 0;

}

public int getResult()

{

return result;

}

}

package test3\_2;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Ignore;

import org.junit.Test;

public class CalculatorTest

{

private static Calculator calculator = new Calculator();

@Before

public void setUp() throws Exception

{

calculator.clear();

}

@After

public void tearDown() throws Exception

{

}

@Test

public void testAdd()

{

calculator.add(2);

calculator.add(3);

assertEquals(5,calculator.getResult());

}

@Test

public void testSubstract()

{

calculator.add(10);

calculator.substract(2);

assertEquals(8,calculator.getResult());

}

@Ignore("Multiply() Not yet implemented")

@Test

public void testMultiply()

{

}

@Test

public void testDivide()

{

calculator.add(8);

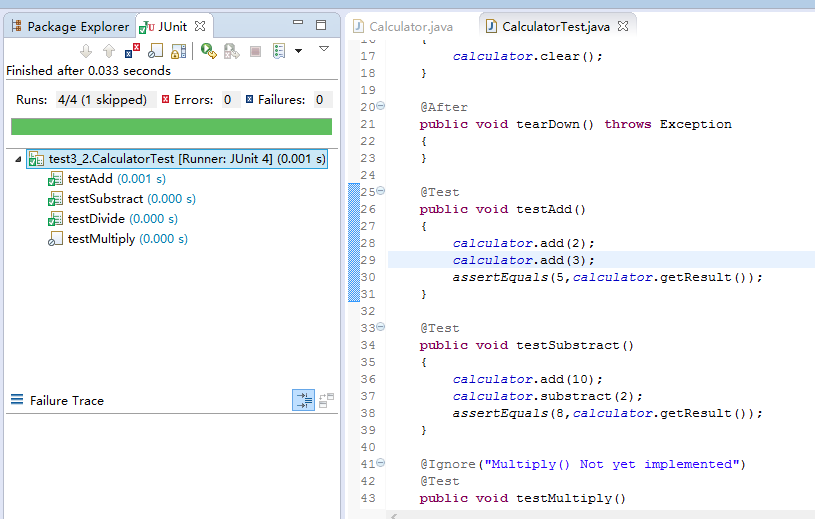
calculator.divide(2);

assertEquals(4,calculator.getResult());

}

}

**运行截图**

****

**四、实验心得**

本次实验，我首次接触到了Junit这个类，在使用下了解了其的功能定位和作用。这个类的包在我Eclipse下都存在，只是没有用到。。实验中Junit的测试只包含了普通函数的测试，现在正好是小学期，为了测试系统中的servlet类，我上网研究了好几种方法，其中一种方法让我调试成功了。据网上的帖子所说，我这种方法用到了Jetty类(查询过程中发现这个类的作用好像不止支持测试)。为什么还要找其他的方法呢，因为servlet中有表单提交过来的Request参数。虽然，通过这个方法，请求参数没问题了，但是没有session中的attribute(因为系统中这个属性是根据登陆而来的)。通过本实验的入门和自己的研究，我对Junit测试有了初步的认识和理解，对我以后编程的规范有很大的帮助。