**Junit单元测试**

**首先一句话：Junit相当于一个集成、自动的main测试方法大全。**

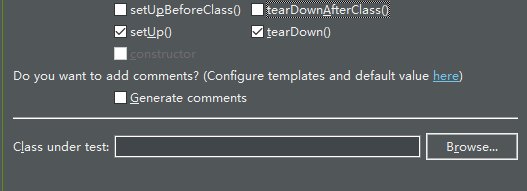
程序开发的时候，调试总是枯燥而且繁琐的，尤其是需要调试的方法多到爆炸的时候，一般来说，等全部代码编写完毕之后再一气调试，会累死人的。在正经的开发作业中，比较流行稳打稳扎的开发流程：**先编写测试用例，再设计程序接口，然后编写类和方法**，测试完毕，确保手上的代码没有问题之后，再一点点加入新的代码。而在测试的技术中，Junit一些自动化的工具可以帮助我们节省很多时间和精力。

下面我们开始利用eclipse来学习如何使用Junit

Eclipse本身自带了Junit，不需要自己再去官网下载。

先编写一个需要调试的类，代码如附件



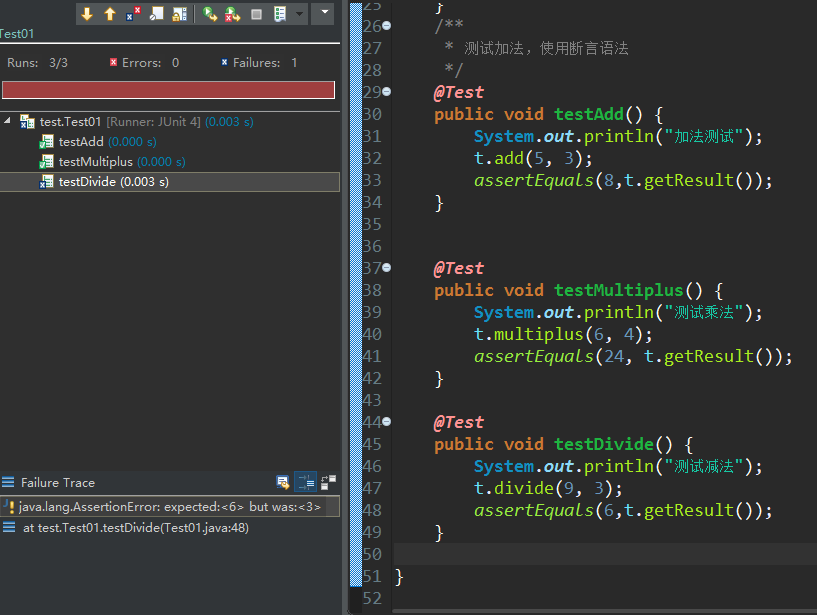
然后通过右键单击src，创建一个专门放置测试用类的包test，右键，创建Junit Test Case，

在class under test中选择我们这次要测试的Target.java，在选择框中选好需要测试的方法，这里，我们选择了add(),Multiplus(),Divide()方法作为测试。

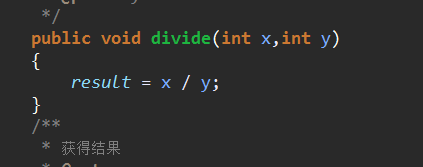
参照附件，使用断言语法编写测试方法，如附件



右键，选择run as JUnit test



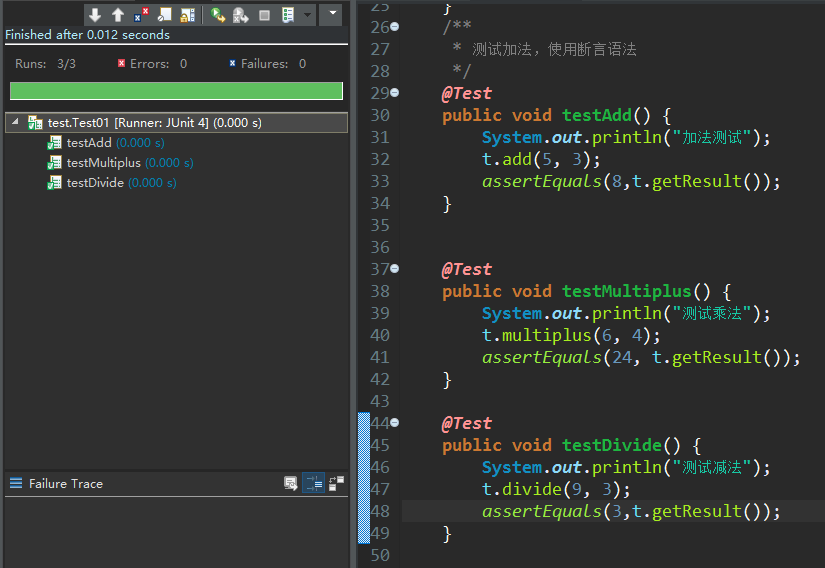
结果发现divide方法测试有误，查看Target的divide方法



**发现是编写方法的人，煞笔了……啊哈哈哈哈哈哈哈哈**

这个例子说明，JUnit只能按照你断言的方法和数据来进行自动化测试，如果你对方法、类等等的要求理解有误，或者仅仅是编写代码时候，业务逻辑有误，JUnit是不能正确地提醒你的。

改正之后，再运行一次，结果如下



好了，最简单的一次JUnit单元测试已经做好。接下来我们学习JUnit的其它用法，可以使你变得更加高效地测试。

1. 参数式测试。

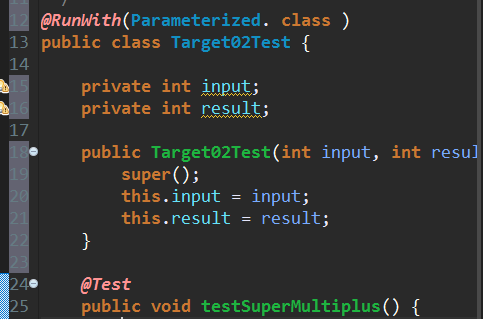
当测试一个方法的时候，往往需要使用不同的边际条件测试很多次，例如测试乘方函数的时候，需要测试0\*0,4\*4，-5\*-5等等不同的情况。这个时候，如果按照一开始的那种方式，每组数据都写一次测试方法，很不划算。

JUnit4开始，支持例如用数组形式设置参数，测试方法接受数组数据用作测试的方法，接下来是例子。

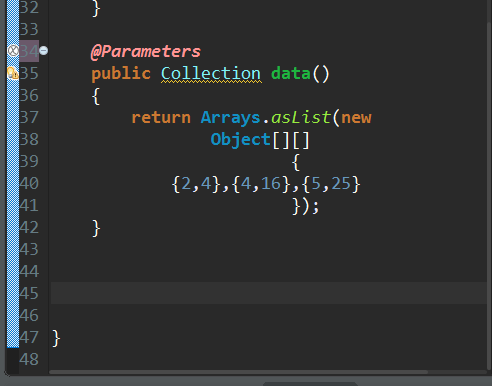
同样，我们先写一个乘方的类Target02



然后，按照之前的方法，创建一个测试类Target02Test，注意，使用参数式测试的时候，需要设置运行器，如图输入并导入相应的包，并且按需要设置两个参数，使用这些参数来构成构造函数。



再写一个返回值是Collection类型的数组方法，用来包含需要测试的数据，注意方法前面要加上@Parameters 如图



**然而，上图是错的！**

@Parameters修饰的参数数组方法必须是static的，所以，后面这个才是对的



这样一次就可以测试3组数据，只需写一个方法，测试结果正确。

1. 一次运行多个测试类（打包测试）

有时候，我们需要将一个个测试类一下子运行，**偷懒啊**！！这个时候就需要@Suite了。需要注意的是，编写SuitTest时，它的方法体里不能有任何方法，他只是其他测试类（或者测试方法）的“集合石”。

同样，他也需要更改默认的运行器，变成@RunWith(Suite.class)，然后在@Suite.SuiteClasses({输入要运行的测试类}),例如

package test;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.junit.runners.Suite;

import org.junit.runners.Suite.SuiteClasses;

@RunWith(Suite.class)

@Suite.SuiteClasses({

Target02Test.class,Test01.class

})

public class TestSuitDemo {

}

结果一下子就测试了两个测试类，如图

