JUnit是一个Java的单元测试框架

Junit 测试也是程序员测试，即所谓的白盒测试，它需要程序员知道被测试的代码如何完成功能，以及完成什么样的功能

右键项目，Build Path🡪Configure Build Path🡪Add library，选中junit

因为Eclipse已经集成了Junit工具，所以不用另外下载。当然在官网下也行

Calculate.java中的方法待测试

calcTest.java:

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class calcTest {

private Calculate c;

@Test

public void testAdd() {

c=new Calculate();

assertEquals(5, c.add(2, 3));

assertEquals(6, c.add(2, 3));

}

@Test

public void testSubstract() {

c=new Calculate();

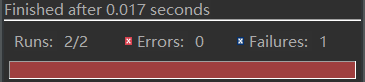
assertEquals(3, c.substract(8, 5));

}

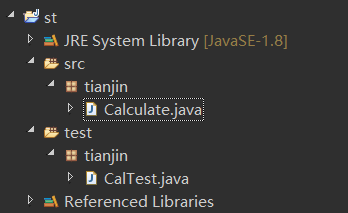
}

鼠标单击testAdd或者testSubstract方法，然后右键🡪run as🡪Junit Test

绿条没有错误

红条有错，在Failure Trace里有具体位置

* 测试方法必须使用public void 进行修饰，不能带任何的参数
* 新建一个源代码目录来存放我们的测试代码，即将测试代码和项目业务代码分开；测试类所在的包名应该和被测试类所在的包名保持一致



如图，建一个和src平行的test的source folder，里边是tianjin包，里边是测试类

请注意：如果test目录下不是tianjin包，代码中必须有一行

import tianjin.Calculate;

来引入src中的Calculate类。如果test目录和src目录一样都是tianjin包，则不需要引入

这是因为，src和test两个文件夹中的java代码都编译在了项目bin目录下，编译成.class文件，"st\bin\tianjin\Calculate.class"，可以看到，和src、test这些source folder的名字没有关系。所以包名一样，编译后的.class文件位置就一样，即使在不同的source folder中。所以不需要import

* 测试单元中的每个方法必须可以独立测试，测试方法间不能有任何的依赖
* 测试类使用Test作为类名的后缀（不是必须，不如CalcTest类），测试方法使用test作为方法名的前缀（不是必须，比如testAdd()方法）

@Test:

a)(expected=XXException.class)如果程序的异常和XXException.class一样，则测试通过

b)(timeout=100)如果程序的执行能在100毫秒之内完成，则测试通过

@Test(timeout = 1000)

public void testLoginTimeout() throws Exception {

loginAction.login(new User("admin", "admin"));

}

@Ignore: 被忽略的测试方法：加上之后，暂时不运行此段代码

@Before: 每一个测试方法之前运行

@After: 每一个测试方法之后运行

@Before

public void testBeforeClass(){

c = new Calculator();

}

@BeforeClass: 方法必须必须要是静态方法（static 声明），所有测试开始之前运行

@AfterClass: 方法必须要是静态方法（static 声明），所有测试结束之后运行

@BeforeClass

public static void beforeClass(){

System.out.println("@BeforeClass");

}

beforeClass在before之前，afterClass在after之后