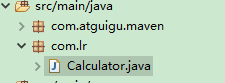
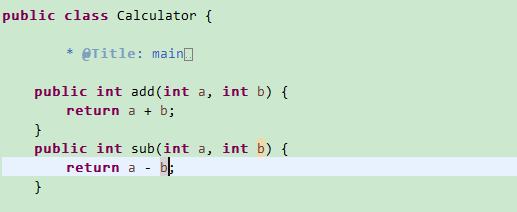
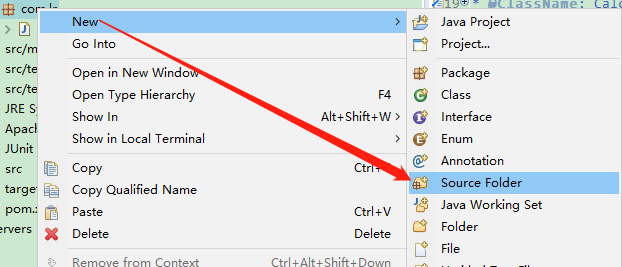
1. 编写一个用于测试的类。



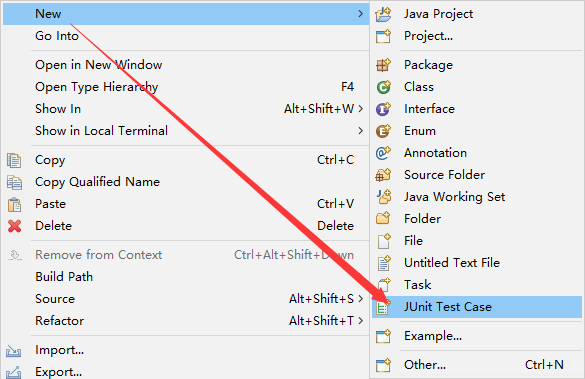


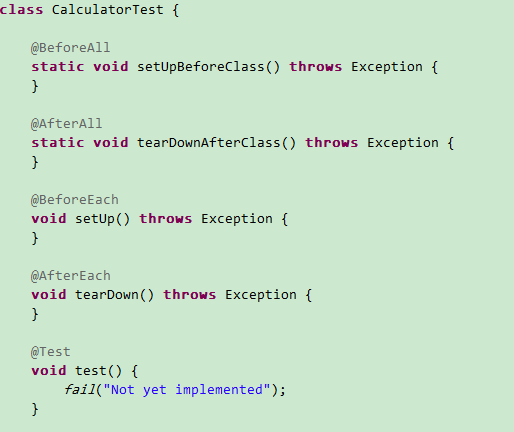
1. 新建测试类

第一步创建一个source folder 资源文件夹 test，资源文件夹和普通的文件夹区别是，资源文件夹中的java文件可以被自动编译。



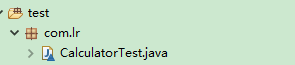
第二步**编写java文件对应的测试类**。

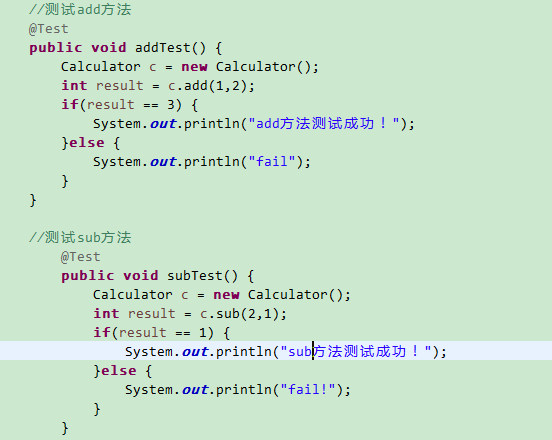




1. 编写测试

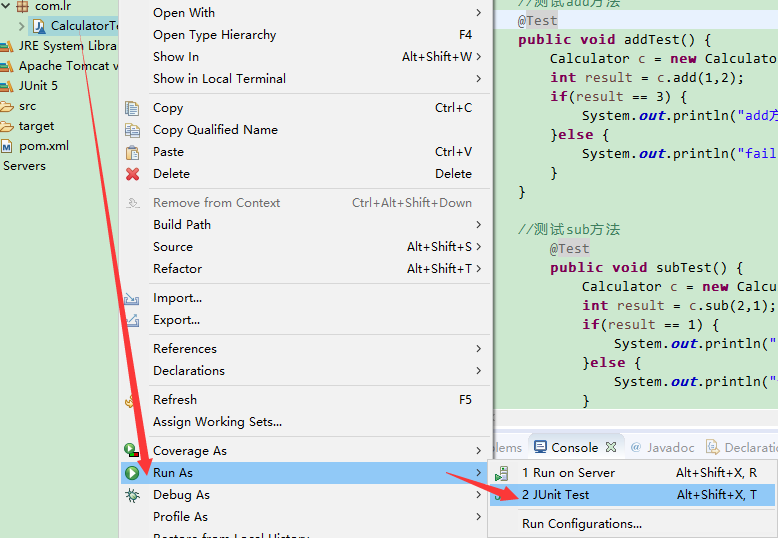
包名要对应。



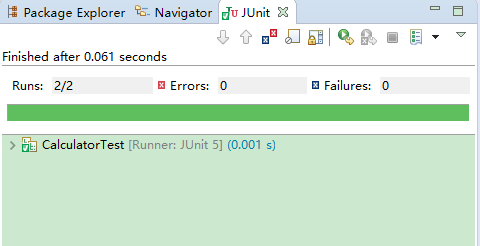


1. 测试结果

点击测试文件右键，如图：



测试结果：



1. 注解说明

**首先介绍 Junit 的几种类似于 @Test 的注解：**

**1.@Test: 测试方法**

**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**

　2.@Ignore: 被忽略的测试方法：加上之后，暂时不运行此段代码

　　3.@Before: 每一个测试方法之前运行

　　4.@After: 每一个测试方法之后运行

　　5.@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，注意区分before，是所有测试方法

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

**注意：编写测试类的原则：**

　　  ①测试方法上必须使用@Test进行修饰

        ②测试方法必须使用public void 进行修饰，不能带任何的参数

        ③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

        ④测试类所在的包名应该和被测试类所在的包名保持一致

        ⑤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

        ⑥测试类使用Test作为类名的后缀（不是必须）

        ⑦测试方法使用test作为方法名的前缀（不是必须）

必须有断言，assert