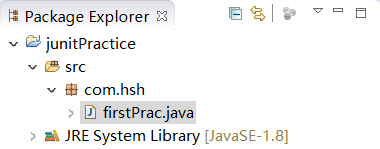
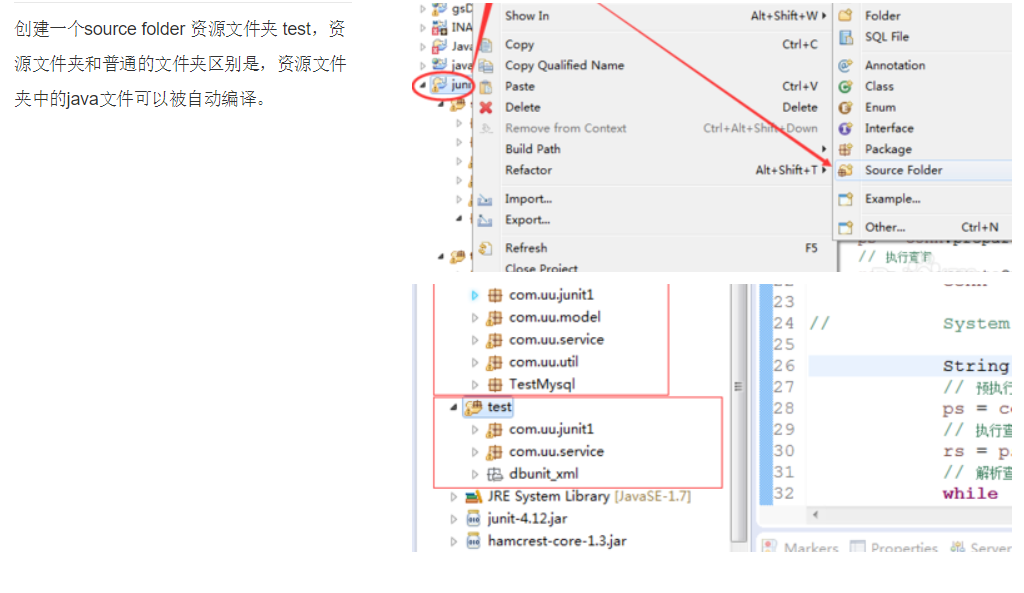
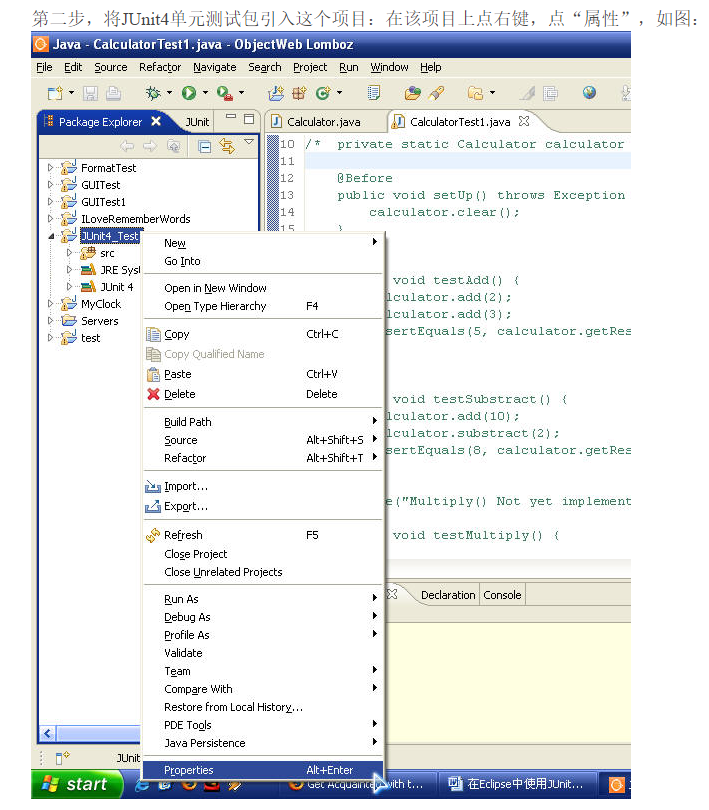
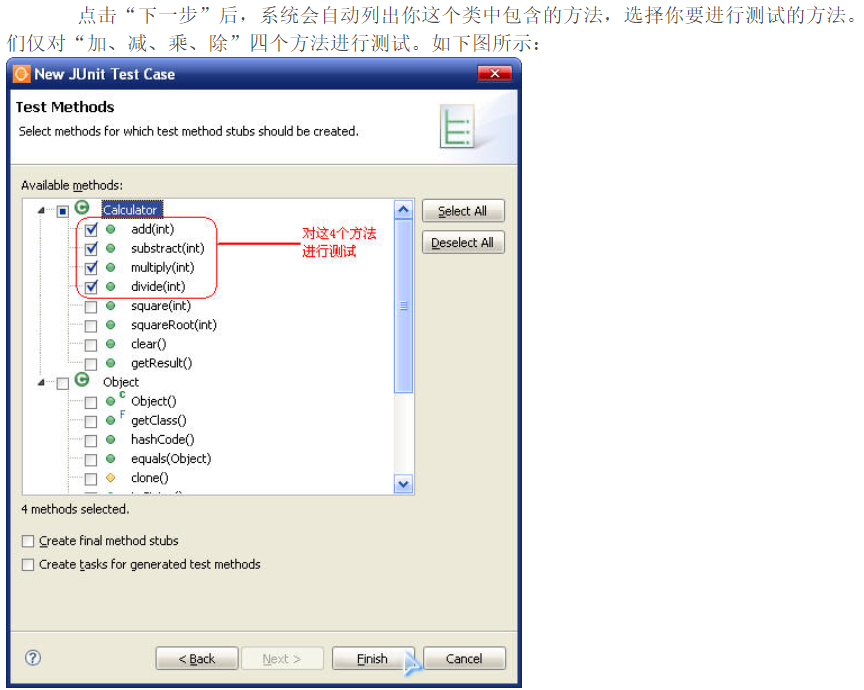
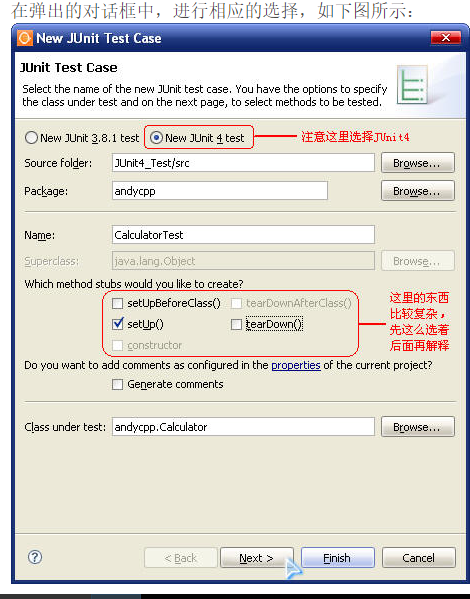
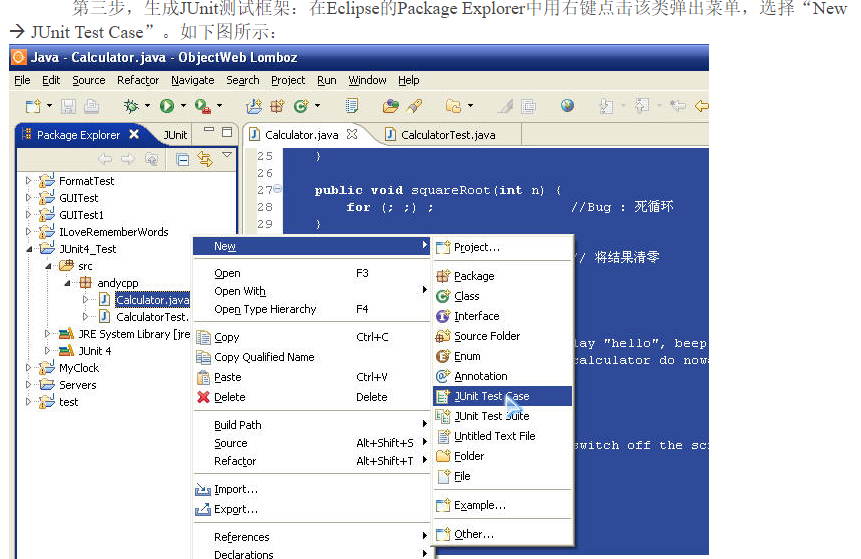
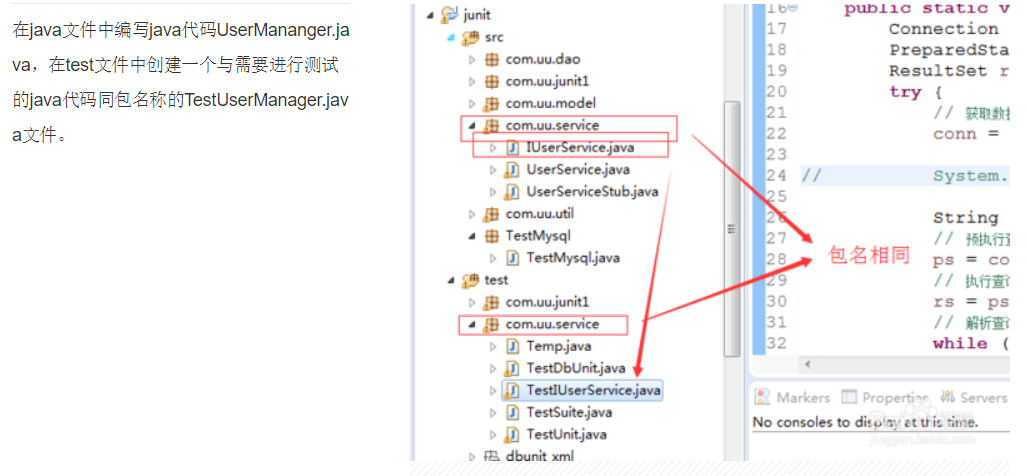
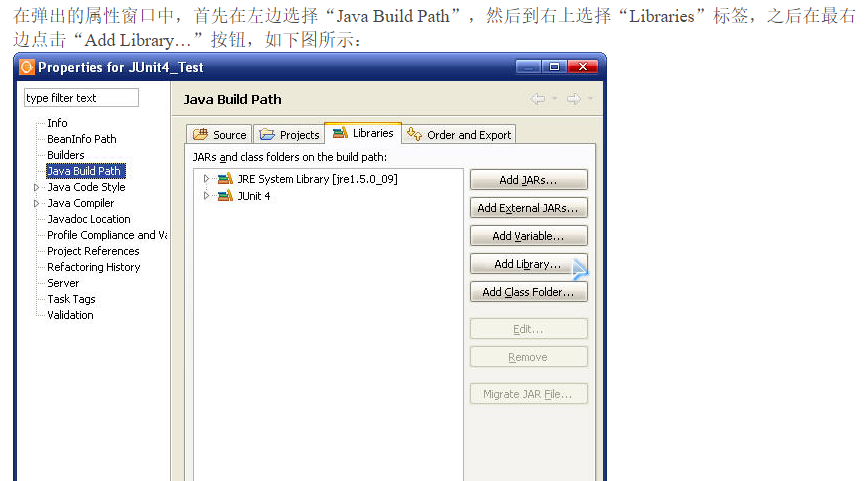
# 初学Junit

建立一个java工程：







junit4采用的是通过注解的方式（在方法的上面加上@）.

@Before表示在所有方法运行前运行的方法;

@After表示在所有的方法运行之后执行的方法;

@Test表示这是一个测试方法

@BeforeClass表示在这个测试类构造之前执行的方法

@AfterClass表示在这个测试类构造之后执行的方法

setUpBeforeClass()方法在整个类初始化之后调用，一般用来做测试准备工作；

tearDownAfterClass()方法在整个类结束之前调用，一般做测试的清理工作。

这里使用了注解，简单的理解为依赖反射，在测试类前标注@Test即可，我看了JUnit3中，写测试方法名必须以testxxx标识，而且需要继承TestCase等等，JUnit4大量的使用注解来解决这一现象，既方便又快捷。

下面介绍一下经常使用到的一些注解：

**@Before：初始化方法**

**@After：释放资源**

**@Test：测试方法，在这里可以测试期望异常和超过时间**

**@Ignore：忽略的测试方法**

**@BeforeClass：针对所有测试方法，只执行一次，且必须为static void**

**@AfterClass：针对所有测试，只执行一次，且必须为static void**

**@RunWith：指定测试类使用某个运行器**

**@Parameters：指定测试类的测试数据集合**

**@Rule：允许灵活添加或重新定义测试类中的每个测试方法的行为**

**@FixMethodOrder:指定测试方法的执行顺序**

可以看看JUnit对话框中出现的四个方法，打钩的是运行成功；“/”是添加@Ignore注解的方法，也就是忽略测试的方法；每个方法后面都有运行的时间，控制台输出方法运行的顺序

**@BeforeClass –> @Test –> @AfterClass**

**如果加上@Before和@After的话，顺序就是这样**

**@BeforeClass –>@Before –>@Test –> @After–> @AfterClass**

再看提示的错误信息，相差0.5，也就是说误差是0.5，然而我故意写的0.05，当然报错了，这里提示的信息很详细。。。；注意代码里的除法测试，我给的参数是10和3，也就是10/3，然后给的期望值是3，无论给多少都有误差（0.3333333333...），所以我在误差里写上0.5，程序运行正确！



