

实验五 3 JUnit 和 NUnit 的自动测试 实验报告

姓名：姜丽

学号：08133341

班级：计算机科学与技术 13-5 班

编写：2015 年 1 月 13 日

目录

1 实验目的.....	1
2 基本要求.....	1
3 实验步骤和实验结果.....	1
3.1 JUnit 对 Java 集成开发工具 Eclipse 进行白盒测试用例设计及自动测试	1
3.1.1 下载 JUnit.jar.zip.....	1
3.1.2 在 Eclipse 中配置 JUnit.....	1
3.2 NUnit 对 .Net 集成开发工具 Visual Studio2008 进行白盒测试用例设计 及自动测试.....	9
3.2.1 下载并安装 NUnit.....	10
3.2.2 下载并安装 Visual Studio 2008.....	12
3.2.3 在 .NET 中应用 NUnit.....	16
4 实验体会.....	30

1 实验目的

学习 NUnit、JUnit 软件测试工具，针对所选系统实现情况，编写测试计划、设计测试用例，掌握软件自动测试方法。

2 基本要求

- (1) 下载、安装 NUnit、JUnit 工具软件，学习软件使用；
- (2) 运用 NUnit、JUnit 工具软件进行白盒测试用例设计及自动测试；
- (3) 针对目标系统，使用各种测试工具的测试过程。
- (4) 学习 .Net、Java 集成开发环境工具中的软件测试、调试。

3 实验步骤和实验结果

3.1 JUnit 对 Java 集成开发工具 Eclipse 进行白盒测试用例设计及自动测试

3.1.1 下载 JUnit.jar.zip

3.1.2 在 Eclipse 中配置 JUnit

(1) 先创建一个工程，名称为 jiangli，写入程序内容，并在这个工程上点击右键，选择 Properties，点击 Java Build Path，Libraries ->Add External JARs...，选择 JUnit.jar.zip 的位置，如下列图所示：

这是一个测试类：

```
package jiangli;

public class Calculate {

    public int sum(int var1, int var2) {
```

```

        System.out.println("相加的值是: " + var1 + " + " + var2);

        return var1 + var2;
    }

```

这个类有一个公共的方法 `sum()`，它得到输入两个整数，将它们相加并返回结果。将要测试这个方法。

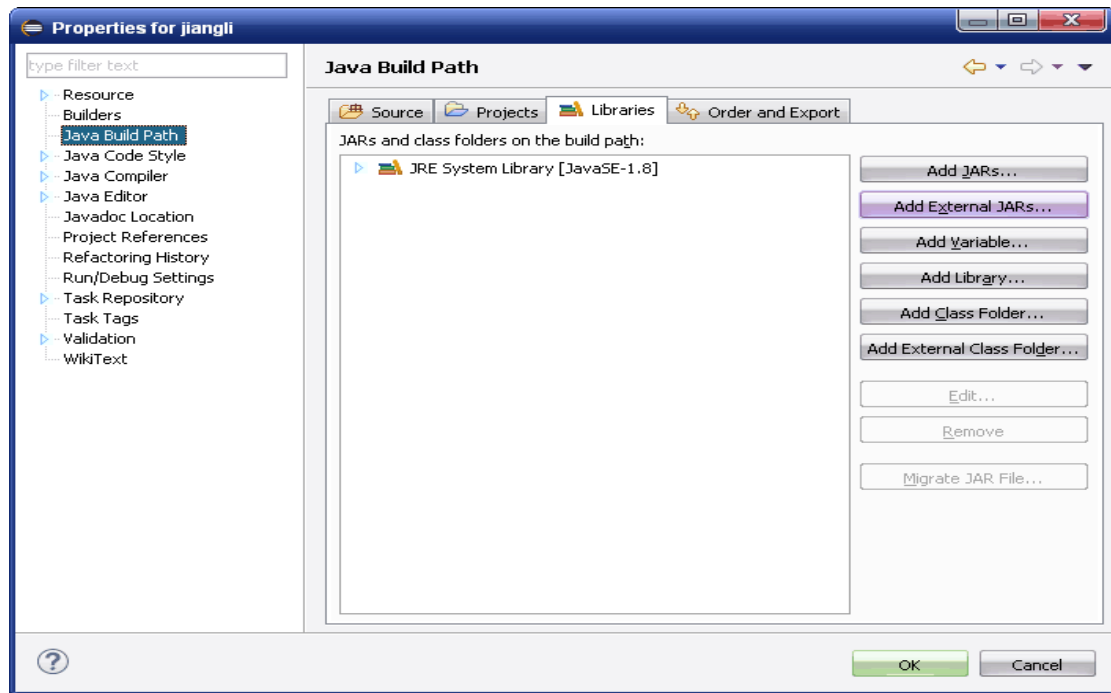


图 3-1 配置 Junit

整个工程的结构如下：

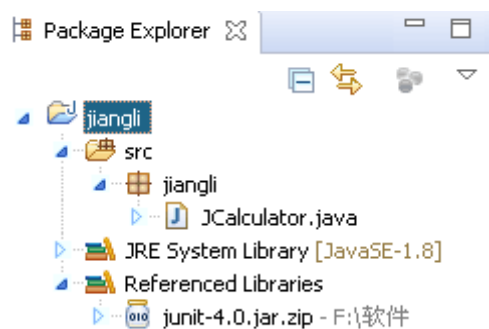


图 3-2 配置完成后的界面

(2) 在需要测试的项目上新建 JUnit Test Case

在需要测试的项目上点击右键选择 New->JUnit Test Case。

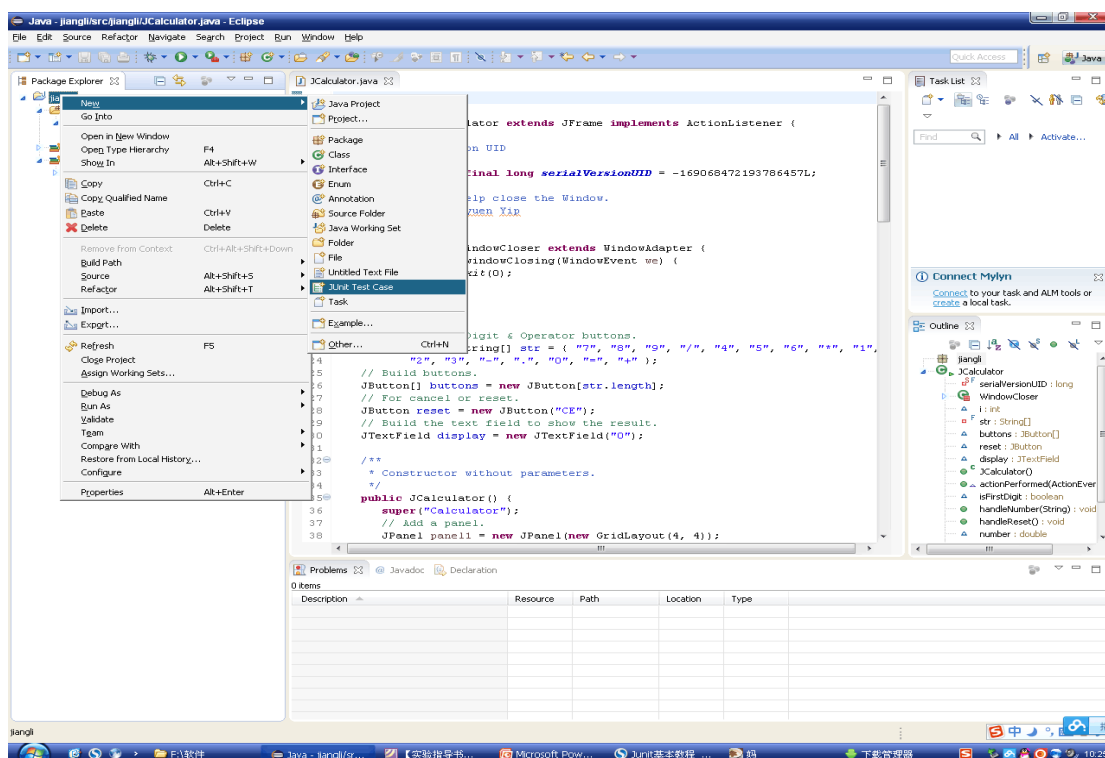


图 3-3 新建 JUnit Test Case 1

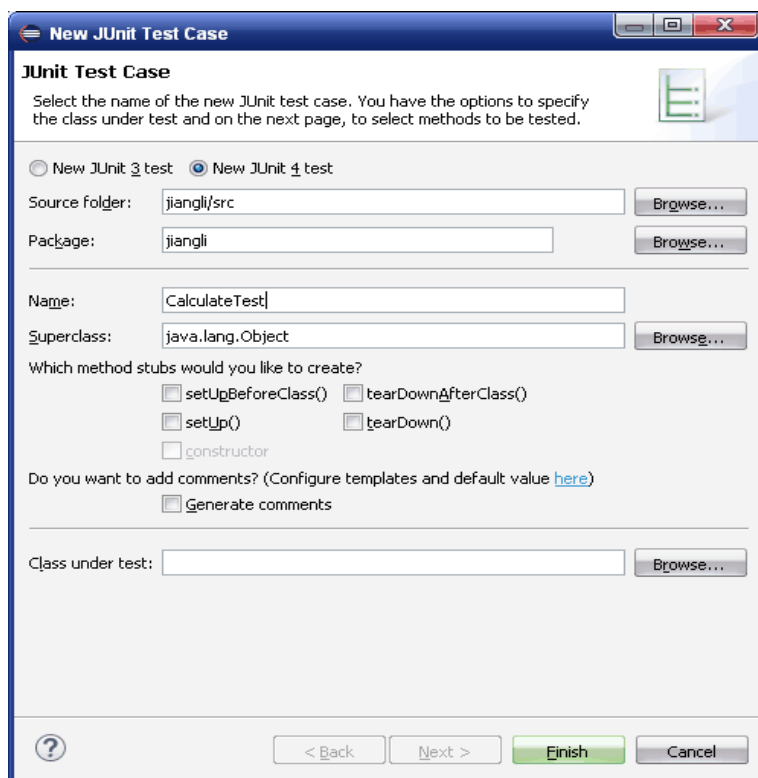


图 3-4 新建 JUnit Test Case 2

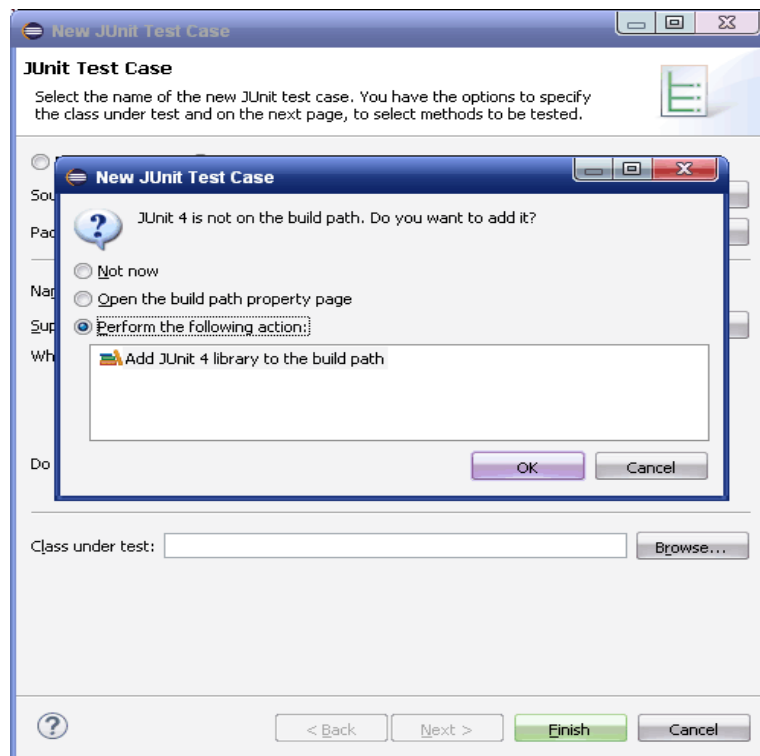


图 3-5 新建 JUnit Test Case 3

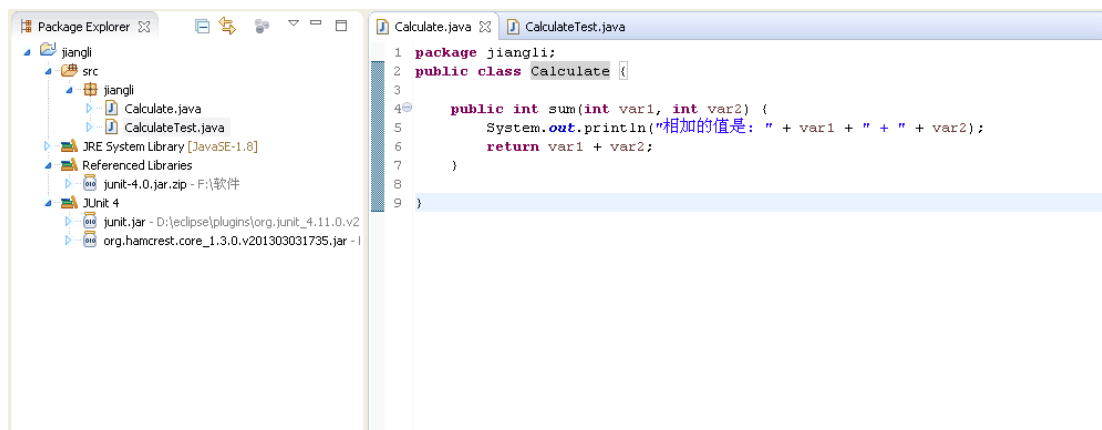


图 3-6 新建 JUnit Test Case 成功

(3) 给 JUnit Test Case 添加测试用例

创建另一个类包括方法，将测试之前的类中的每一个方法。当创建测试用例的详细信息在下面的部分说明。

```
package jiangli;
```

```
import static org.junit.Assert.*;
```

```
import org.junit.Test;
```

```

public class CalculateTest {

    Calculate calculation = new Calculate();

    int sum = calculation.sum(2, 5);

    int testSum = 7;

    @Test

    public void testSum() {

        System.out.println("@Test sum(): " + sum + " = " +
testSum);

        assertEquals(sum, testSum);

    }

}

```

其中，testSum() 方法用于测试公开方法 sum()，方法 assertEquals ([String message], object expected, object actual) 有两个对象作为输入，测试说明这两个对象相等。

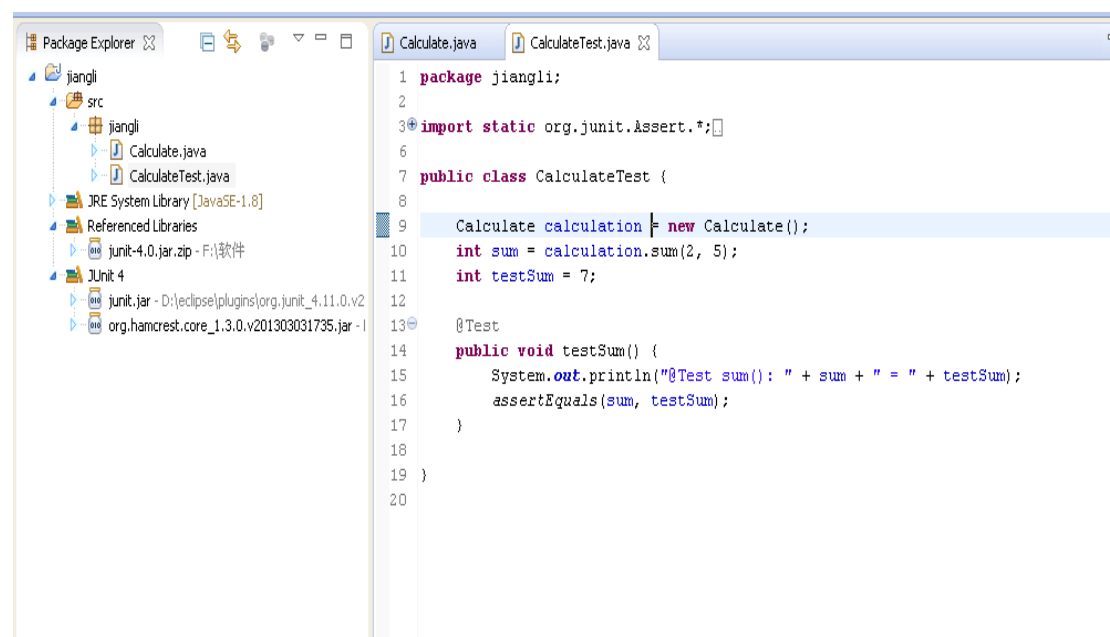


图 3-7 给 JUnit Test Case 添加测试用例

(4) 运行测试用例

在测试用例 Calculate Test 上点击右键，选择 Run As->JUnit Test，运行测试用例。

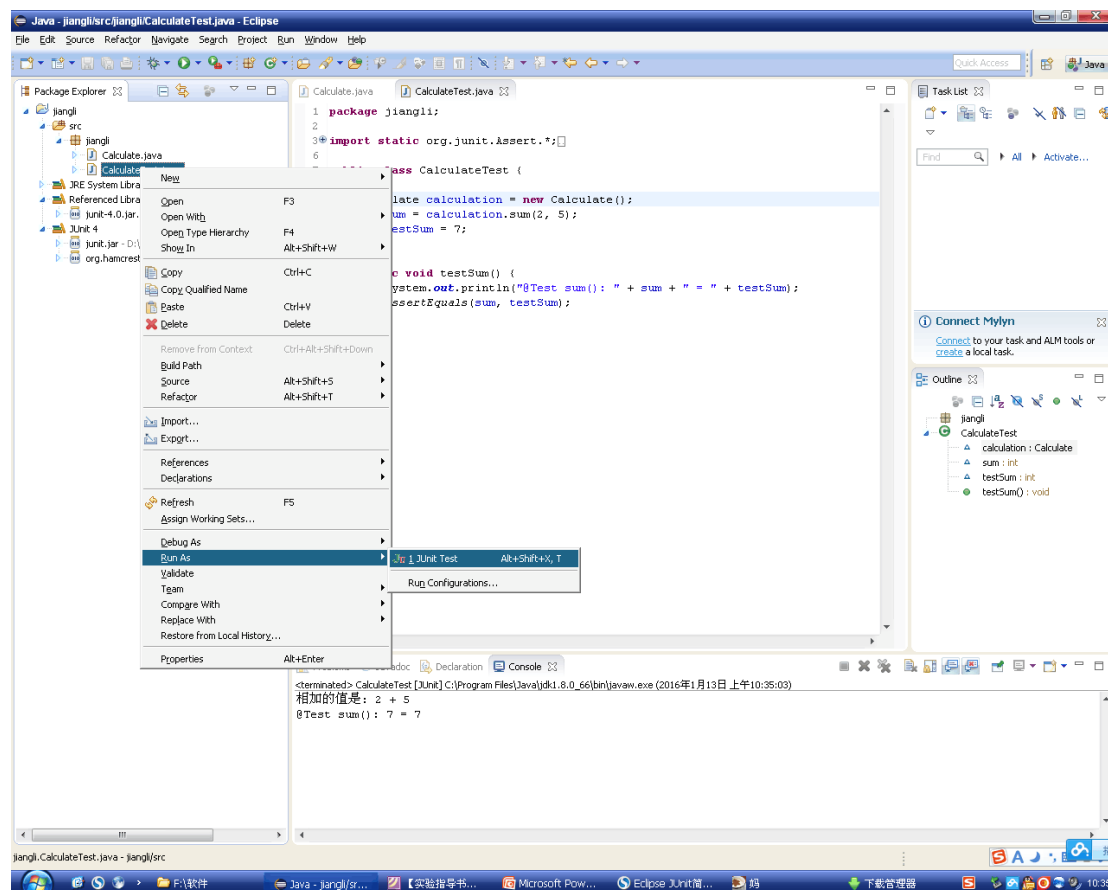


图 3-8 运行测试用例

要查看 JUnit 测试的实际结果，Eclipse IDE 提供了 JUnit 的窗口，它显示了测试的结果。可以在下面的图片中看到(Errors 0)，这种情况下测试成功，JUnit 窗口不显示任何错误或失败。

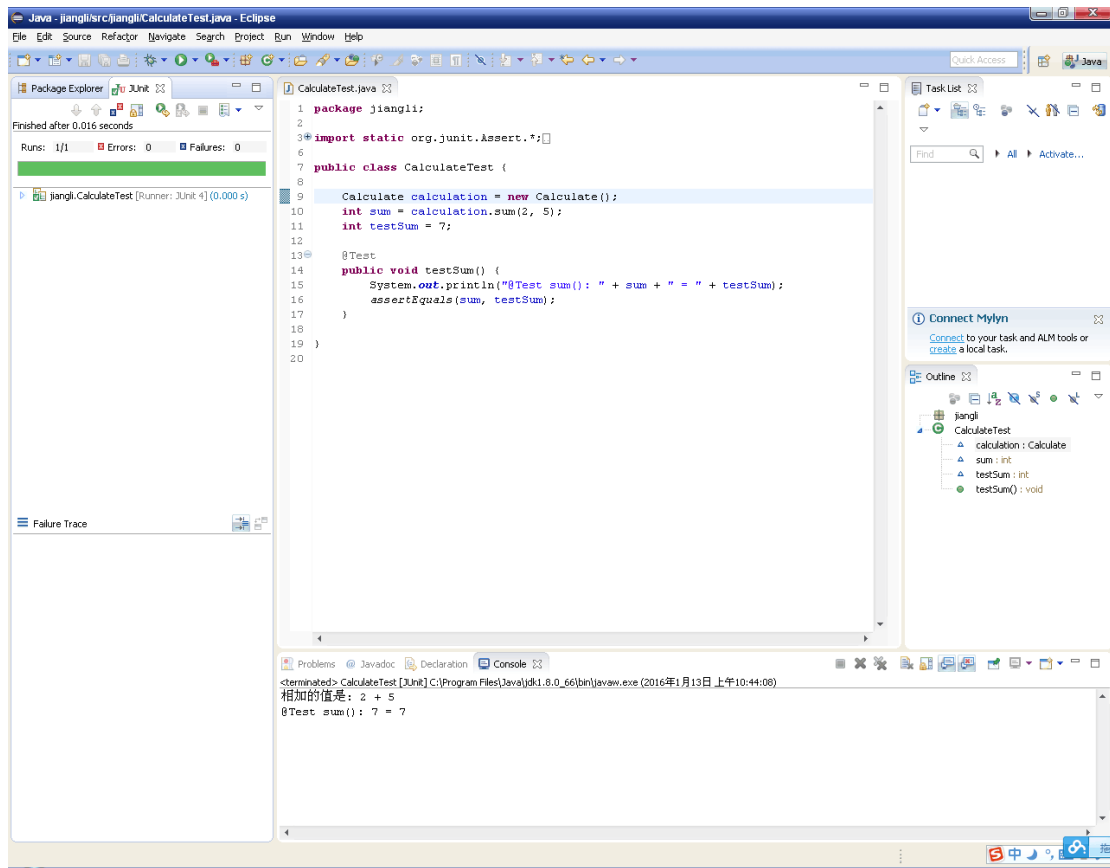


图 3-9 测试用例运行成功

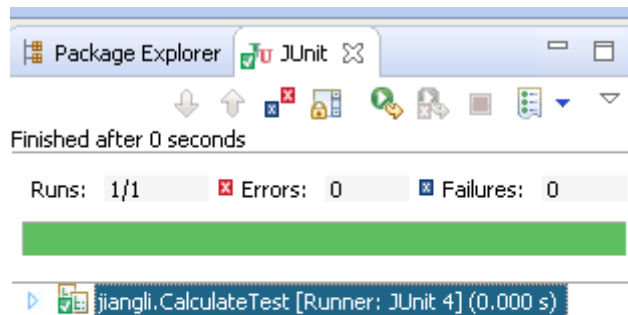


图 3-10 运行成功

(5) 修改测试用例

修改测试用例，使之出错，运行测试用例，查看运行结果。

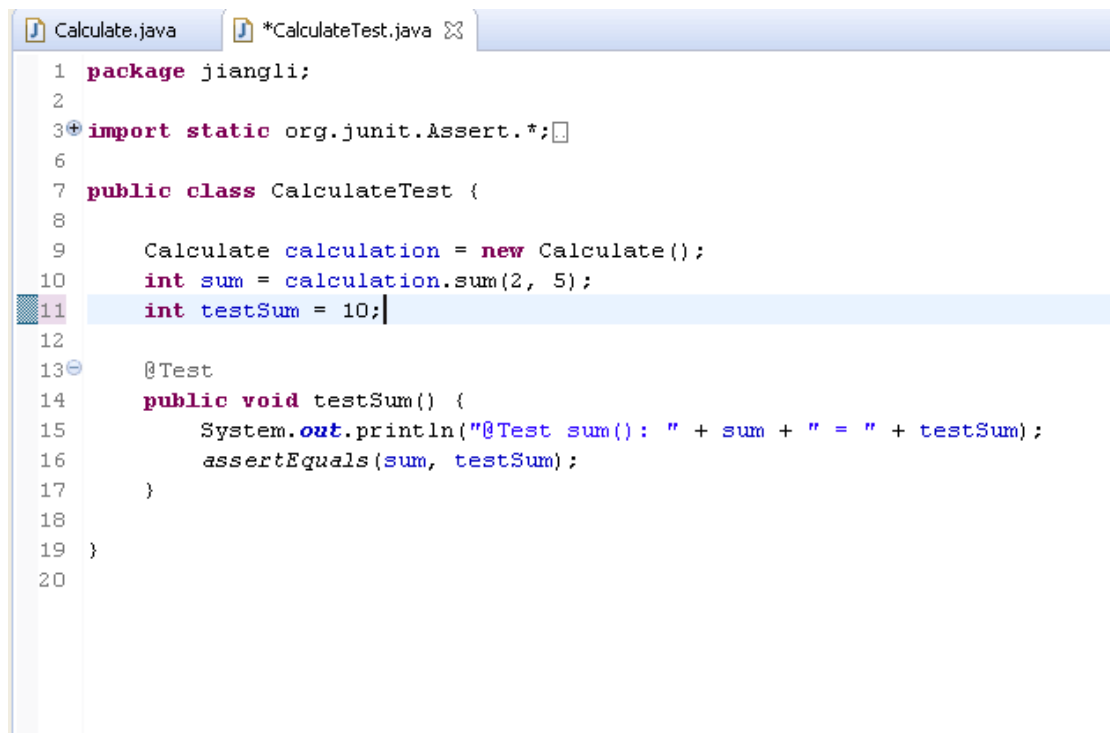


图 3-11 修改测试用例

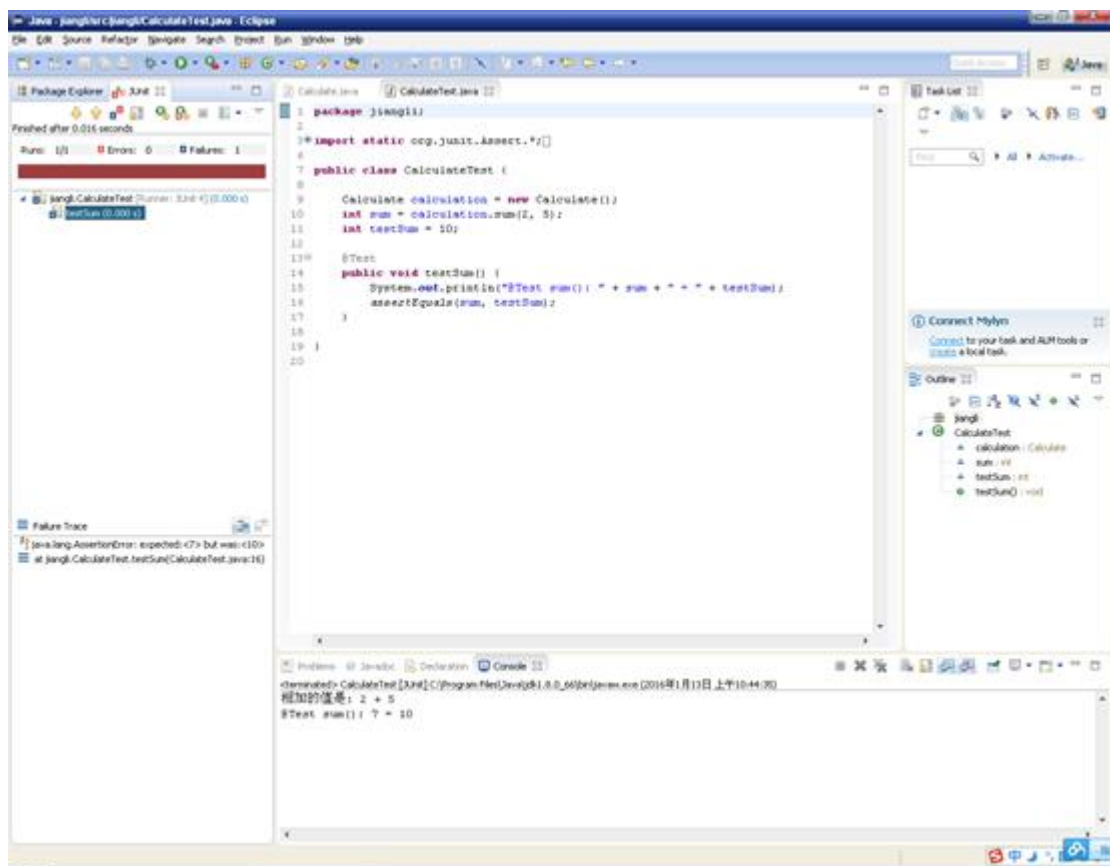


图 3-12 测试出错 1

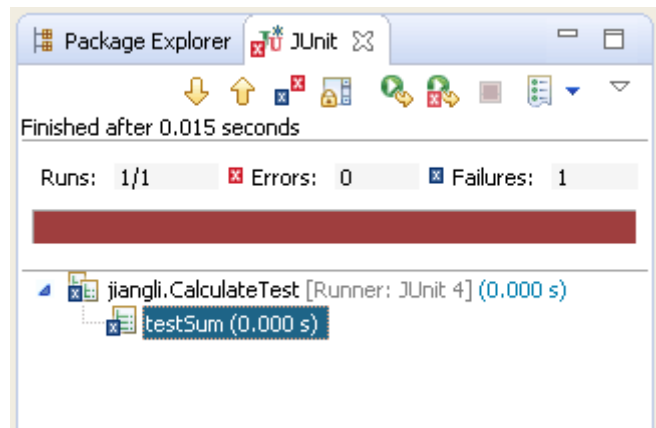


图 3-13 测试出错 2

```
<terminated> CalculateTest [JUnit] C:\Program Files\Java\jdk1.8.0_66\bin\javaw.exe (2016年1月13日 上午10:40:06)
相加的值是: 2 + 5
@Test sum(): 7 = 10
```

图 3-14 测试出错 3

在 JUnit 窗口，有一个错误将出现，并且会显示这样的信息：

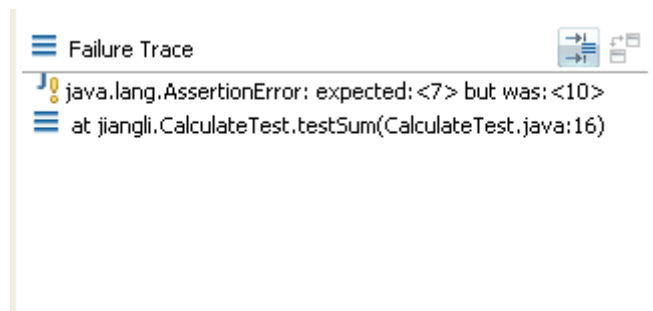


图 3-15 测试出错 4

3.2 NUnit 对 .Net 集成开发工具 Visual Studio2008 进行白盒测试用例设计及自动测试

说明：Visual Studio 2008 支持 .NET Framework 3.5(包括 .NET 2.0 和 3.0)

3.2.1 下载并安装 NUnit

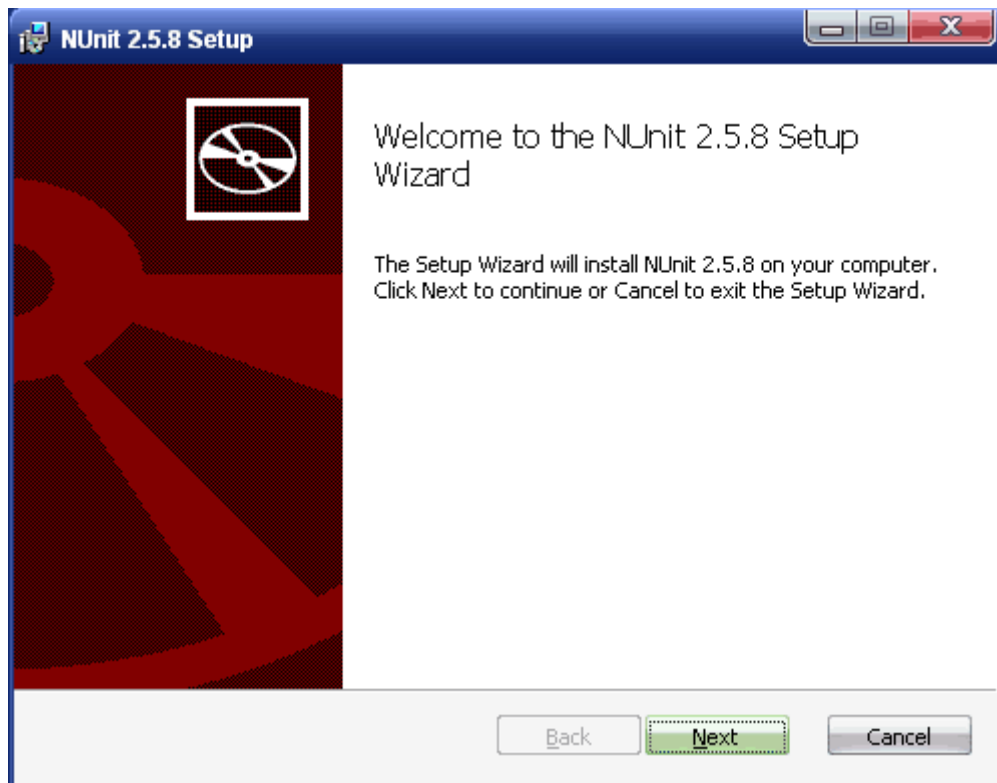


图 3-1 6 安装 NUnit 1

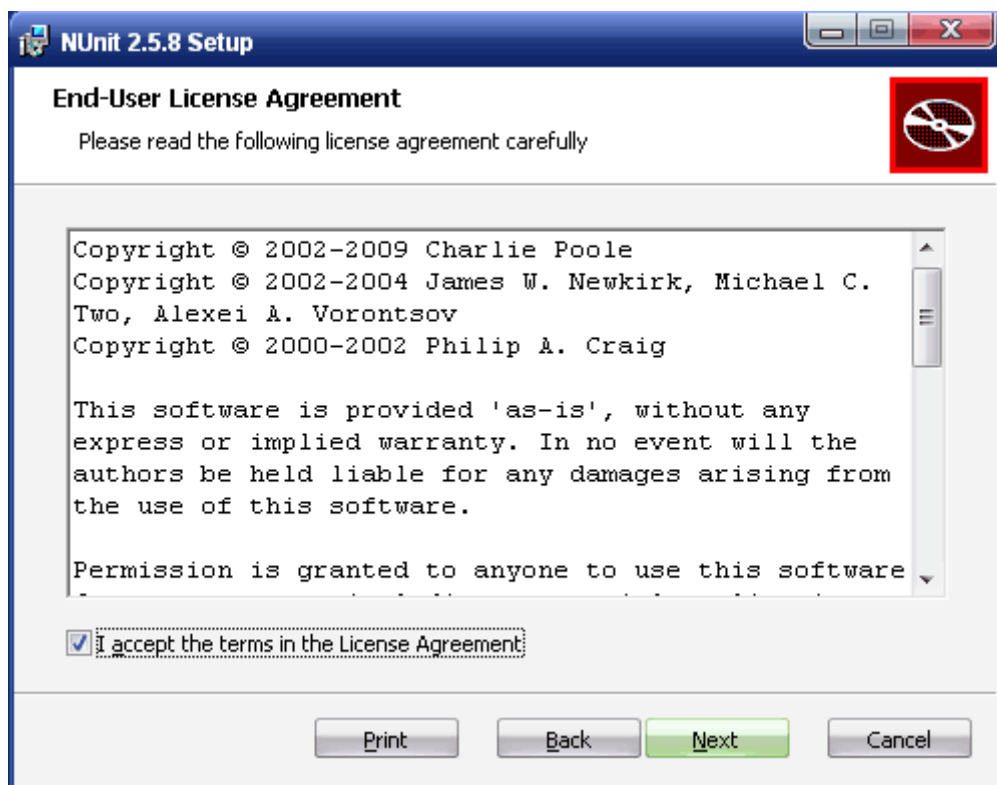


图 3-17 安装 NUnit 1

点击 Typical

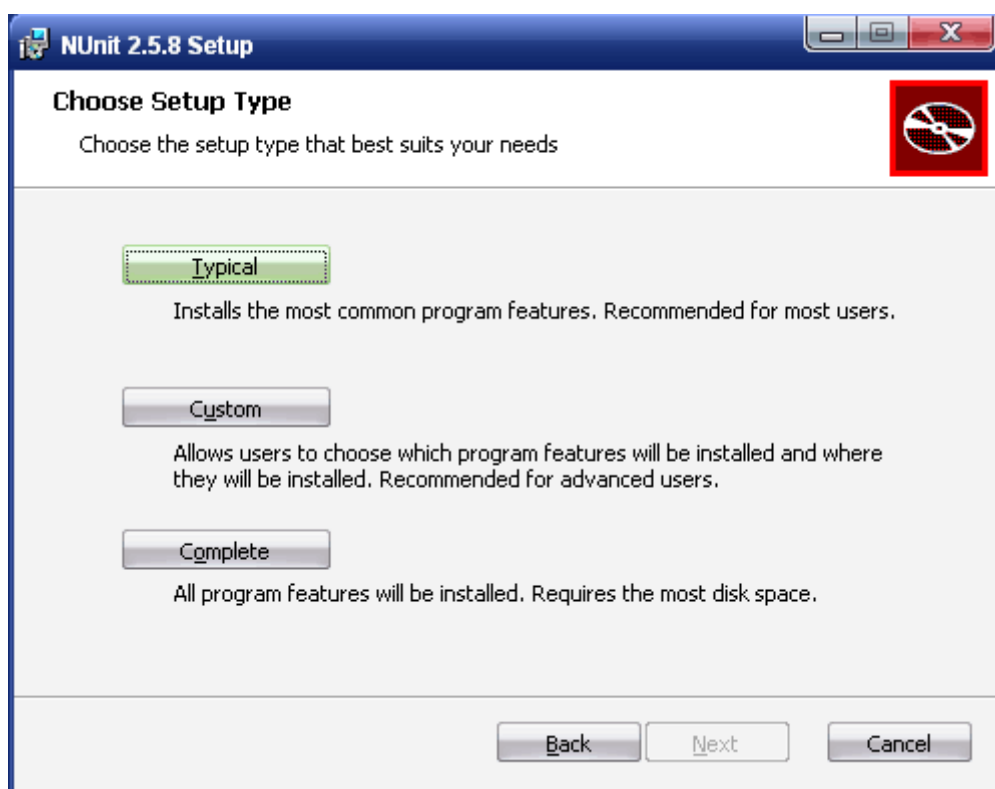


图 3-18 安装 NUnit 3

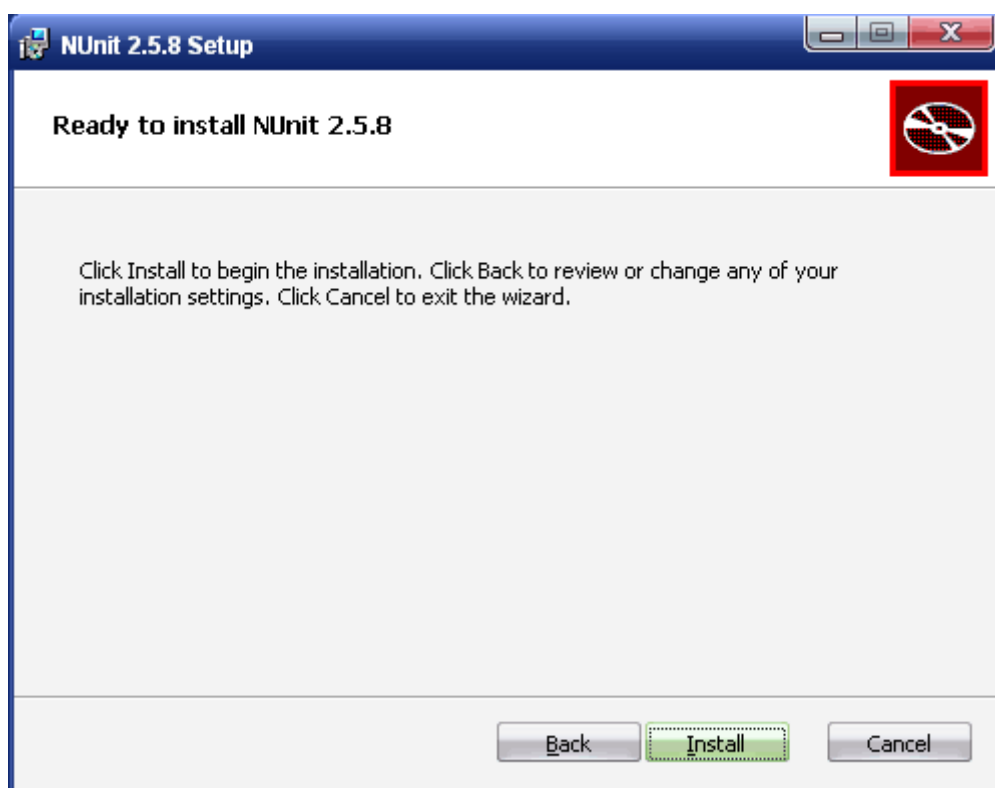


图 3-19 安装 NUnit 4

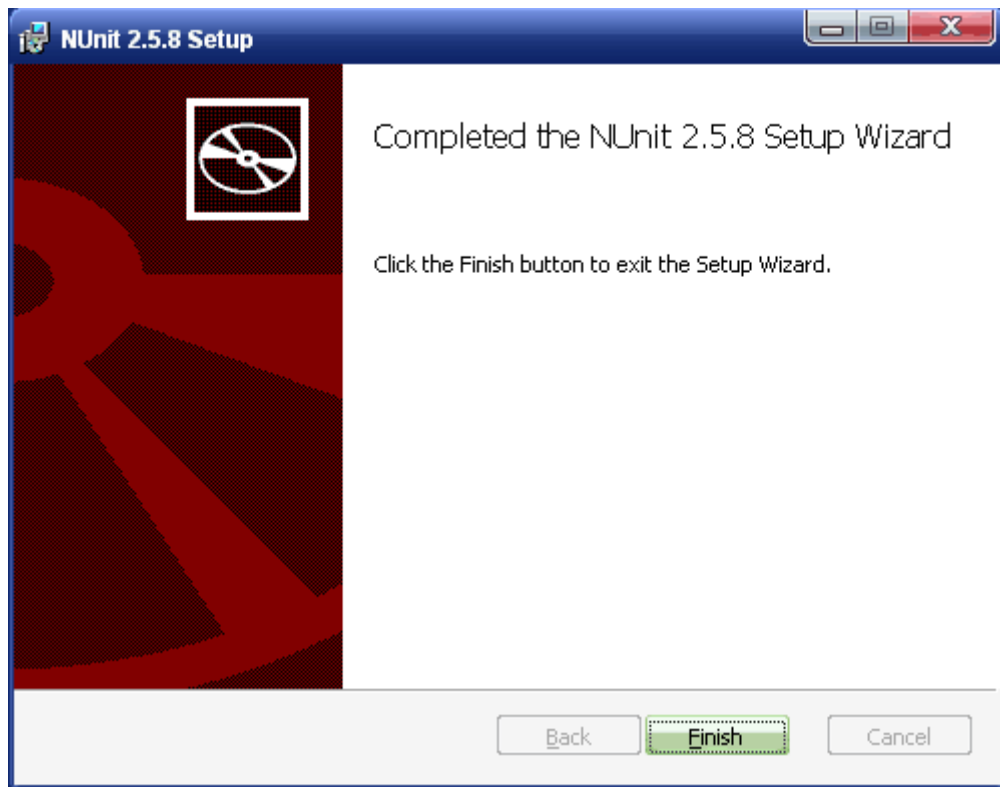


图 3-20 安装成功

3.2.2 下载并安装 Visual Studio 2008



图 3-21 安装 Visual Studio 2008 1



图 3-22 安装 Visual Studio 2008 2

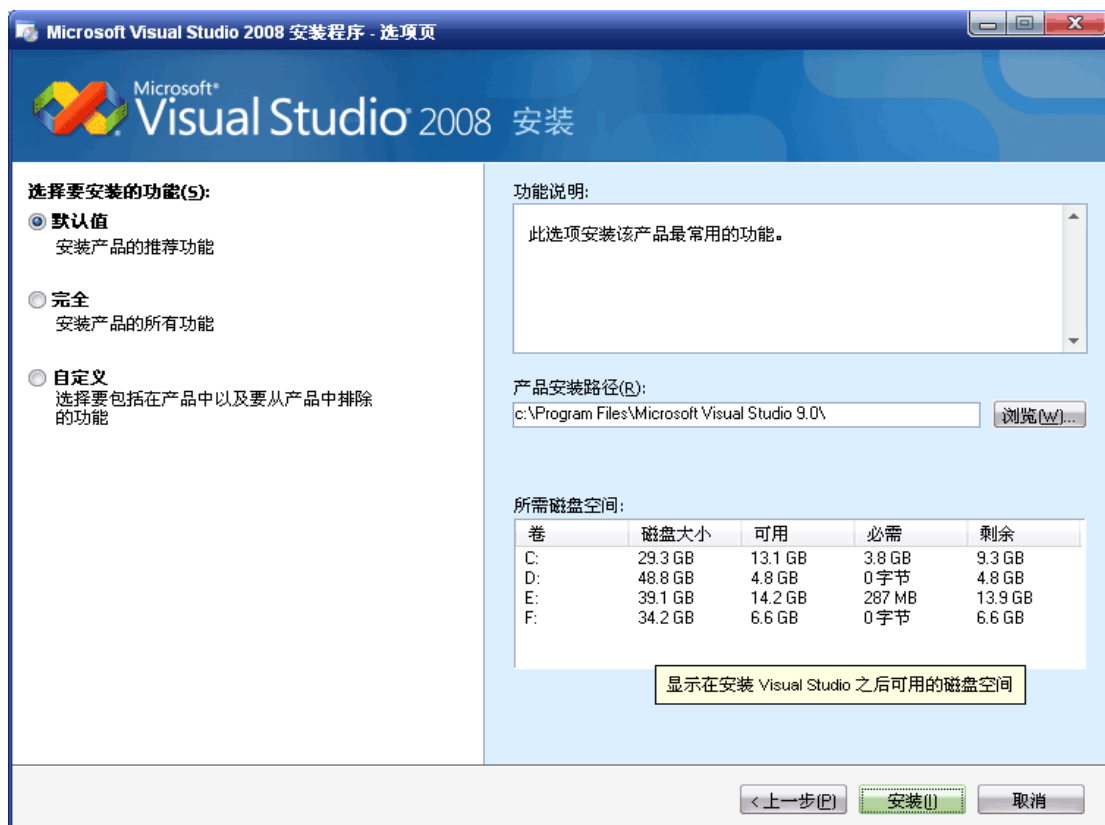


图 3-23 安装 Visual Studio 2008 3



图 3-24 安装 Visual Studio 2008 4



图 3-25 安装 Visual Studio 2008 5



图 3-26 安装 Visual Studio 2008 6

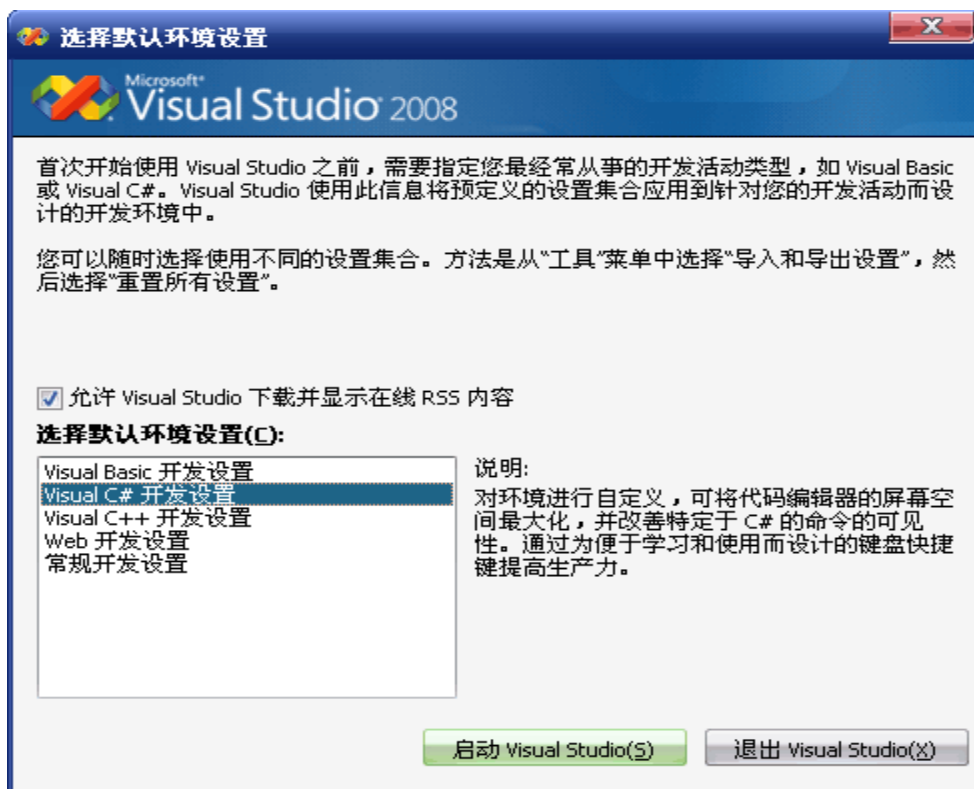


图 3-27 安装 Visual Studio 2008 7



图 3-28 安装 Visual Studio 2008 8



图 3-29 安装 Visual Studio 2008 9

3.2.3 在 .NET 中应用 NUnit

(1) 为测试代码创建一个 Visual Studio 工程

在 Microsoft Visual Studio 中创建一个新的工程，选择文件->新建->新建项目，选择 Visual C#工程作为工程类型，选择“类库”作为模板。将工程命名为 NUnitQuickStart。

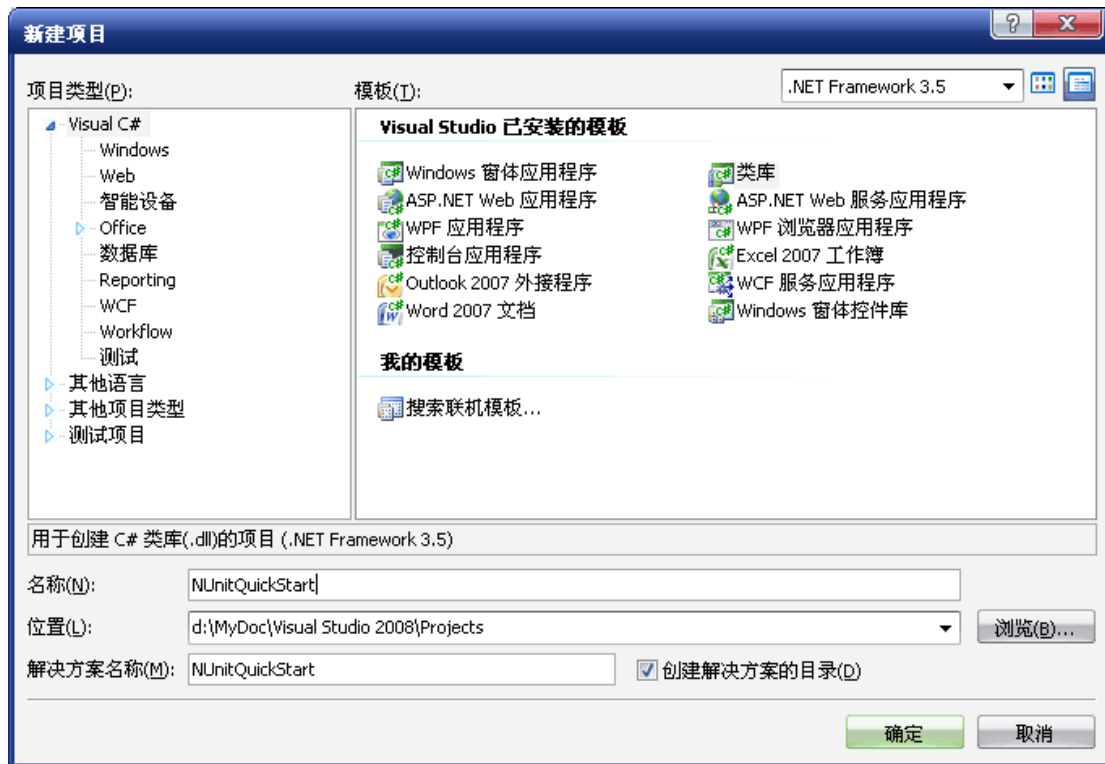


图 3-30 添加项目

(2) 增加一个 NUnit 框架引用

在 Microsoft Visual Studio 中创建用例前，需要增加一个 `nunit.framework` 引用，如下：

在项目 `NUnitQuickStart` 上右击选择引用，然后选择增加引用，在添加引用对话框中找到并选择 `nunit.framework` 组件，点击确定按钮。

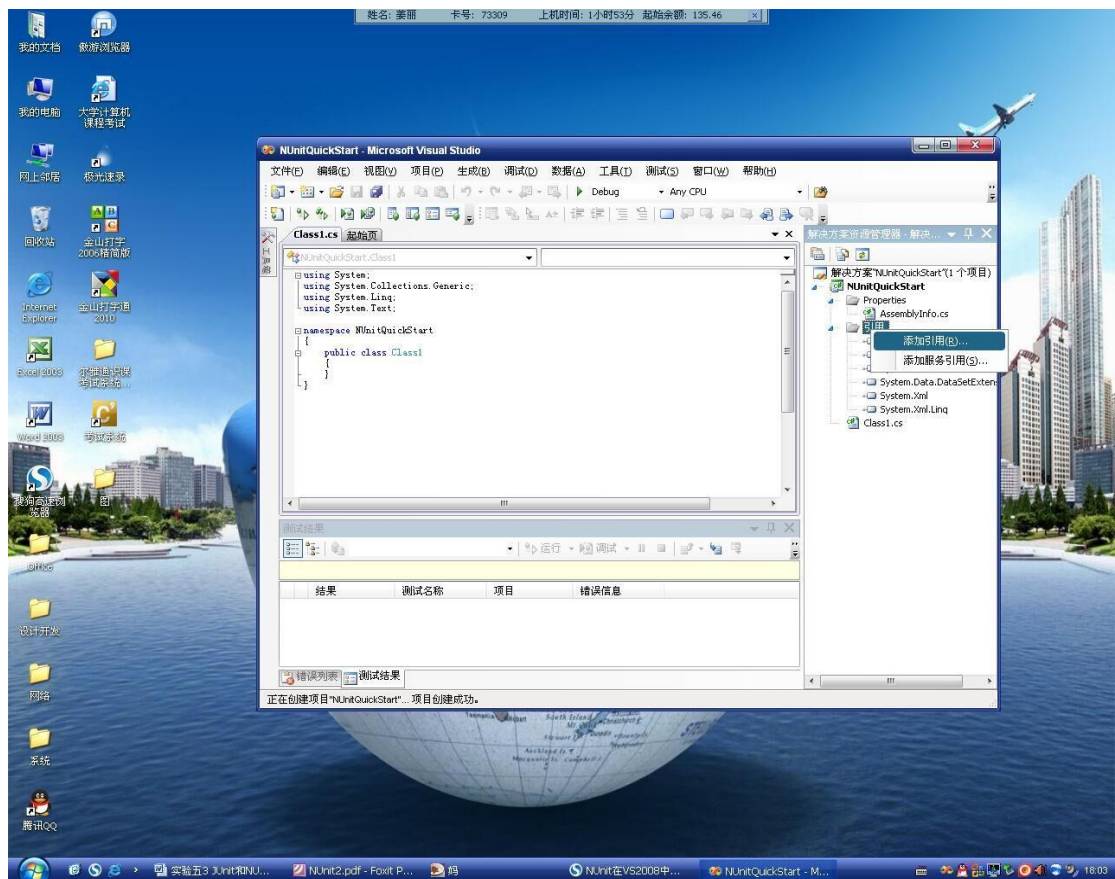


图 3-31 添加引用 1

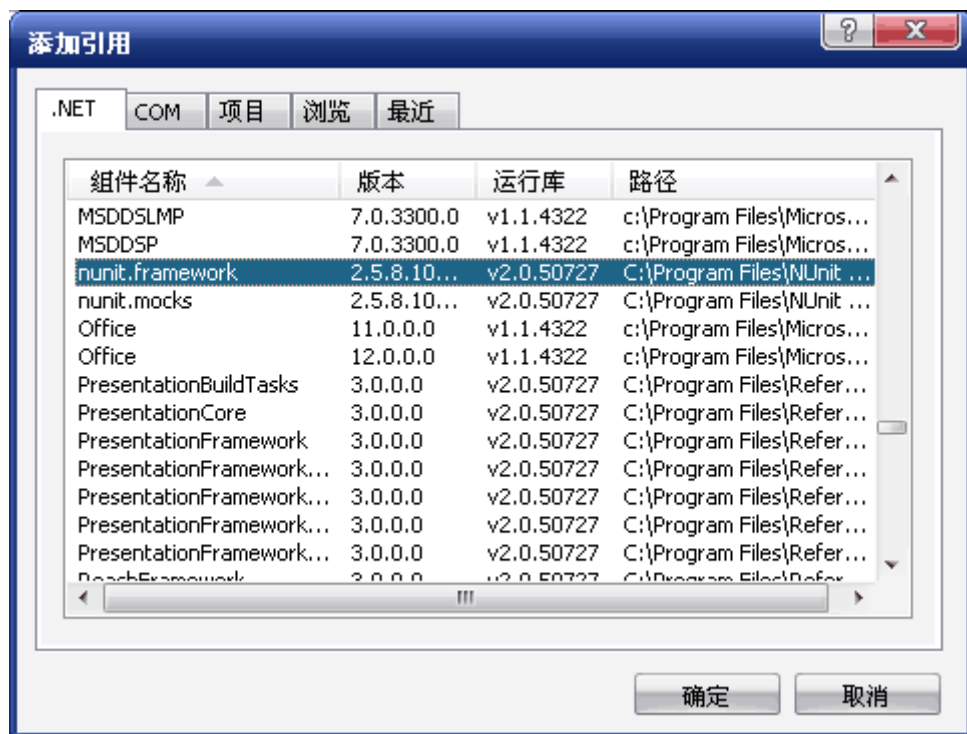


图 3-32 添加引用 2

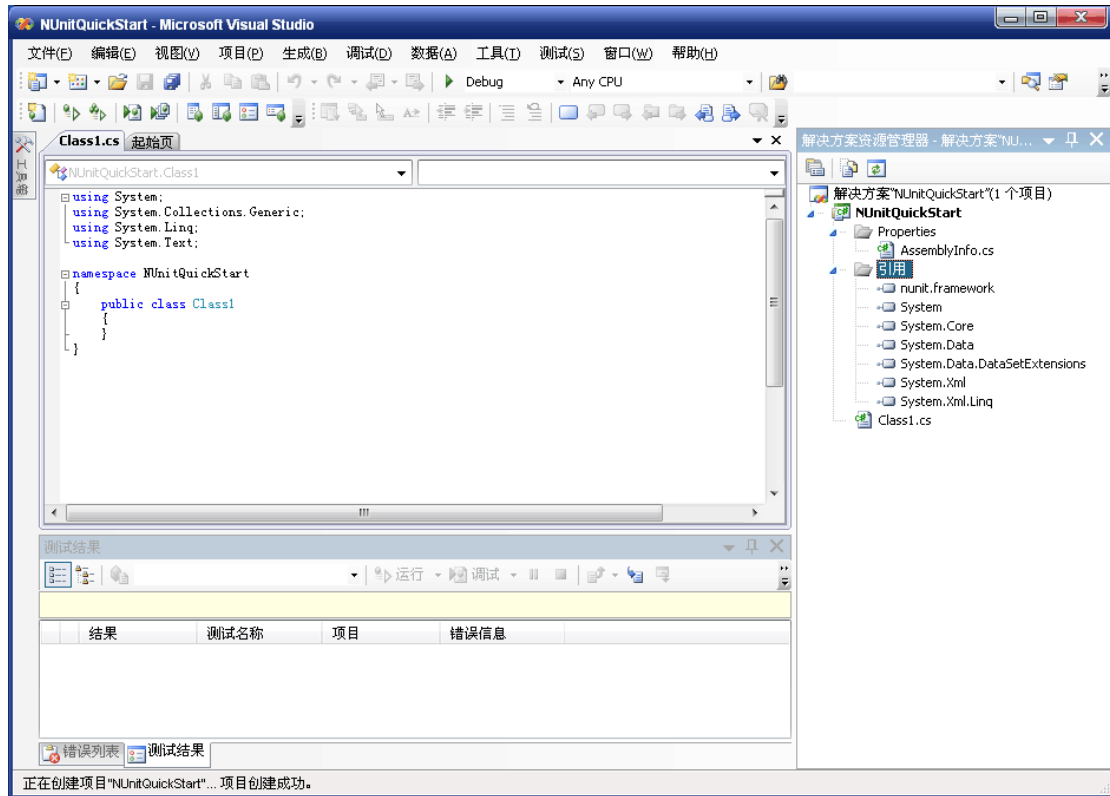


图 3-33 添加引用成功 1

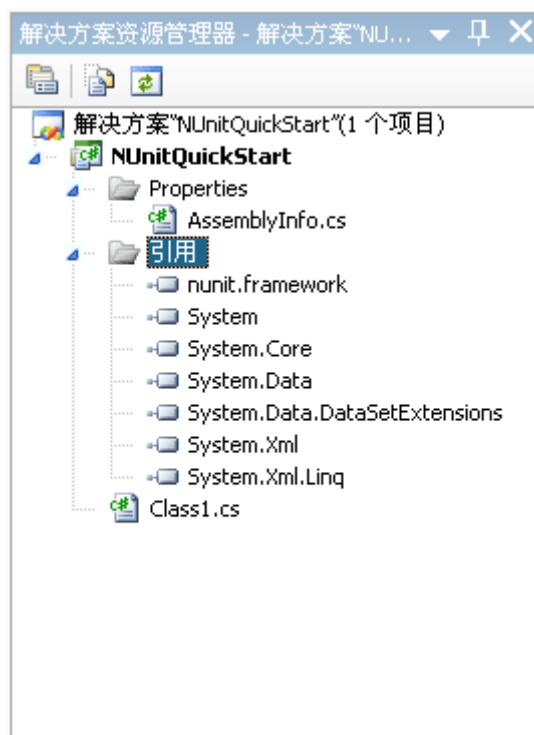


图 3-34 添加引用成功 2

(3) 为工程增加一个类

为工程增加一个 NumbersFixture 类。

```
using System;
using NUnit.Framework;
namespace NUnitQuickStart
{
    [TestFixture]
    public class NumberFixture
    {
        [Test]
        public void AddTwoNumbers()
        {
            int a = 1;
            int b = 2;
            int sum = a + b;
            Assert.AreEqual(sum, 3);
        }
    }
}
```

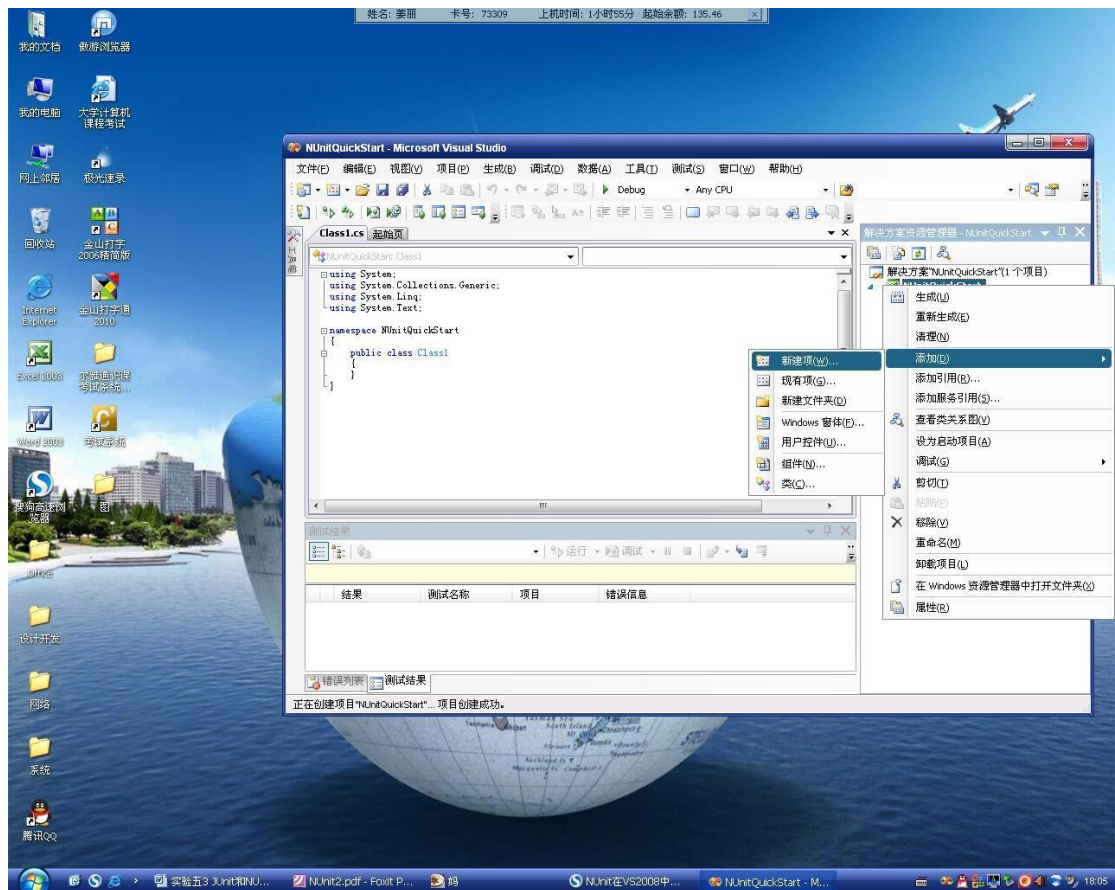



图 3-35 新建类 1



图 3-36 新建类 2

新建成功后，写入测试用例。

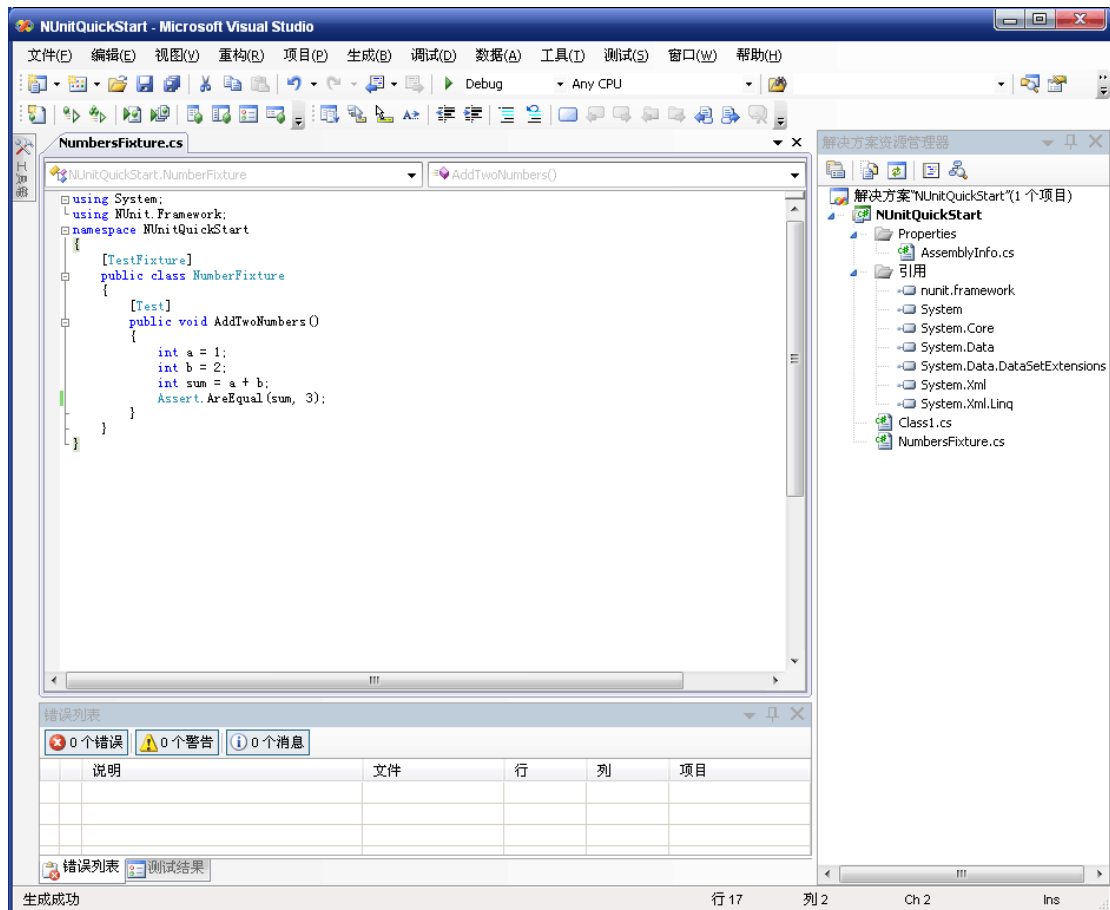


图 3-37 编写测试代码

(4) 建立 Visual Studio 工程后，使用 NUnit 进行测试

为了能在 Visual Studio 中自动运行 NUnit，需要建立 NUnit 作为启动程序：

右击 NUnitQuickStart 工程，在弹出的菜单里选择属性。在显示的对话框的左面点击调试，选择启动操作下的启动外部程序。点击后边的……选择 UNnit 的安装地址，即 `nunit.exe` 所在的位置。

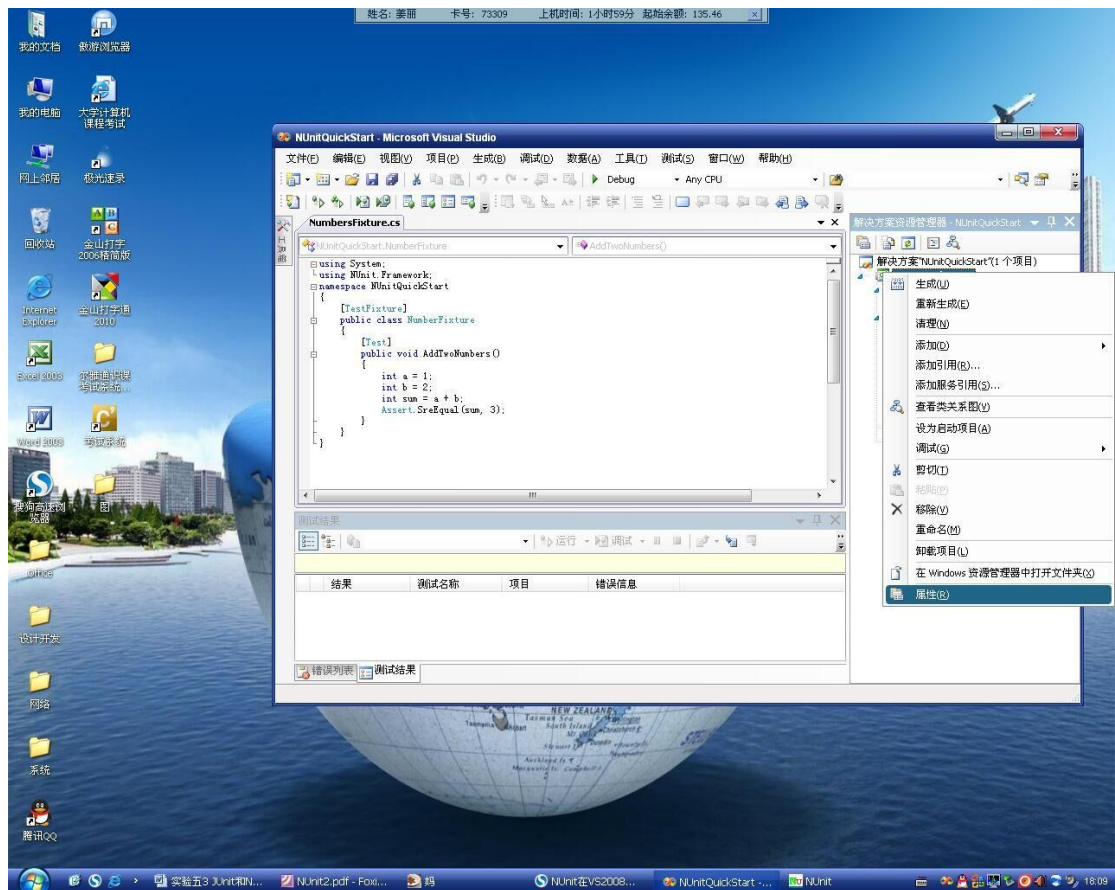


图 3-38 设置 NUnit 为调试程序 1

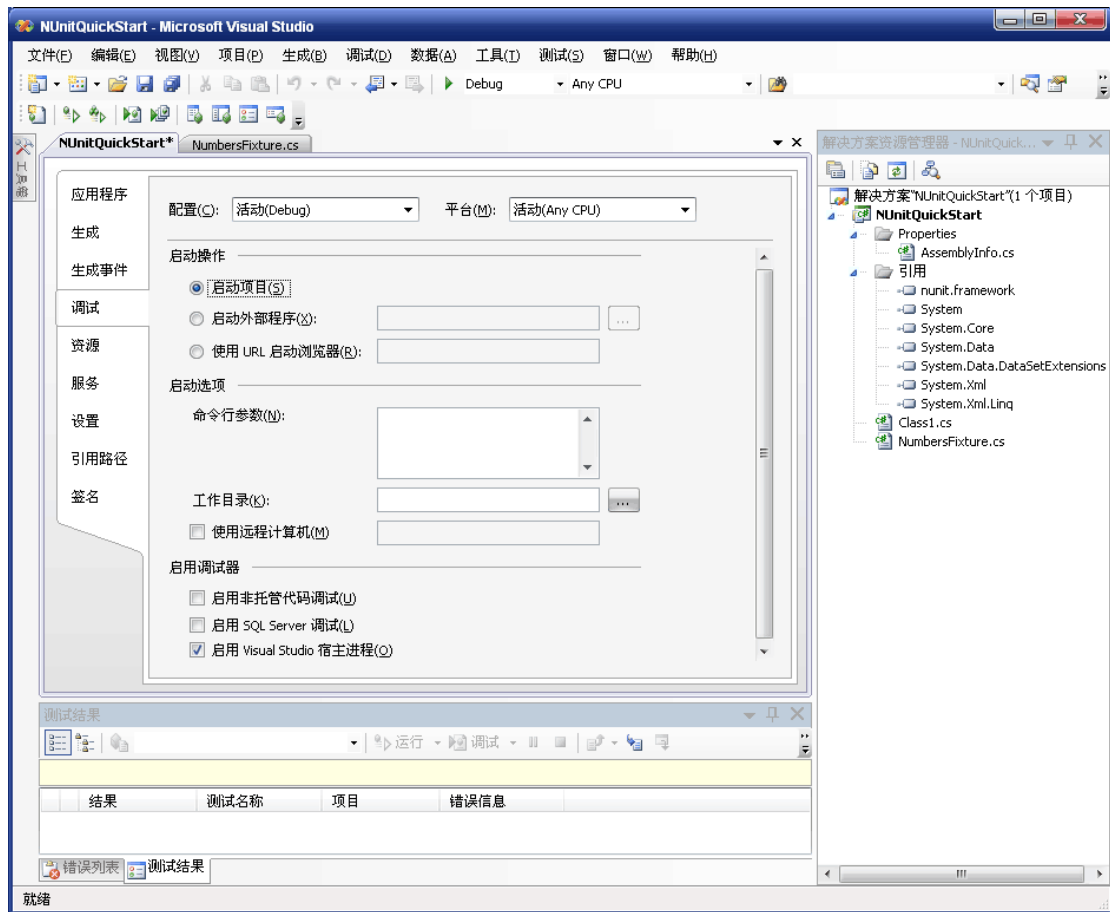


图 3-39 设置 NUnit 为调试程序 2

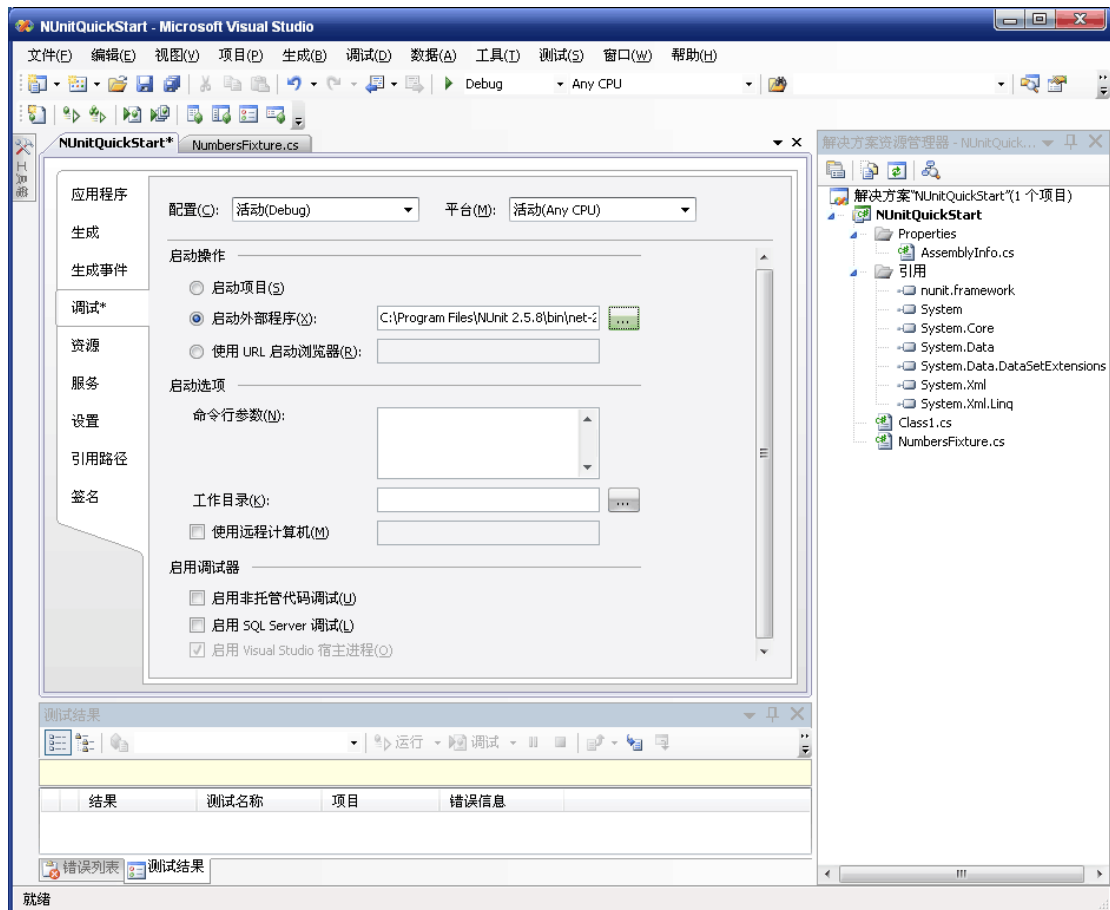


图 3-40 设置 NUnit 为调试程序 3

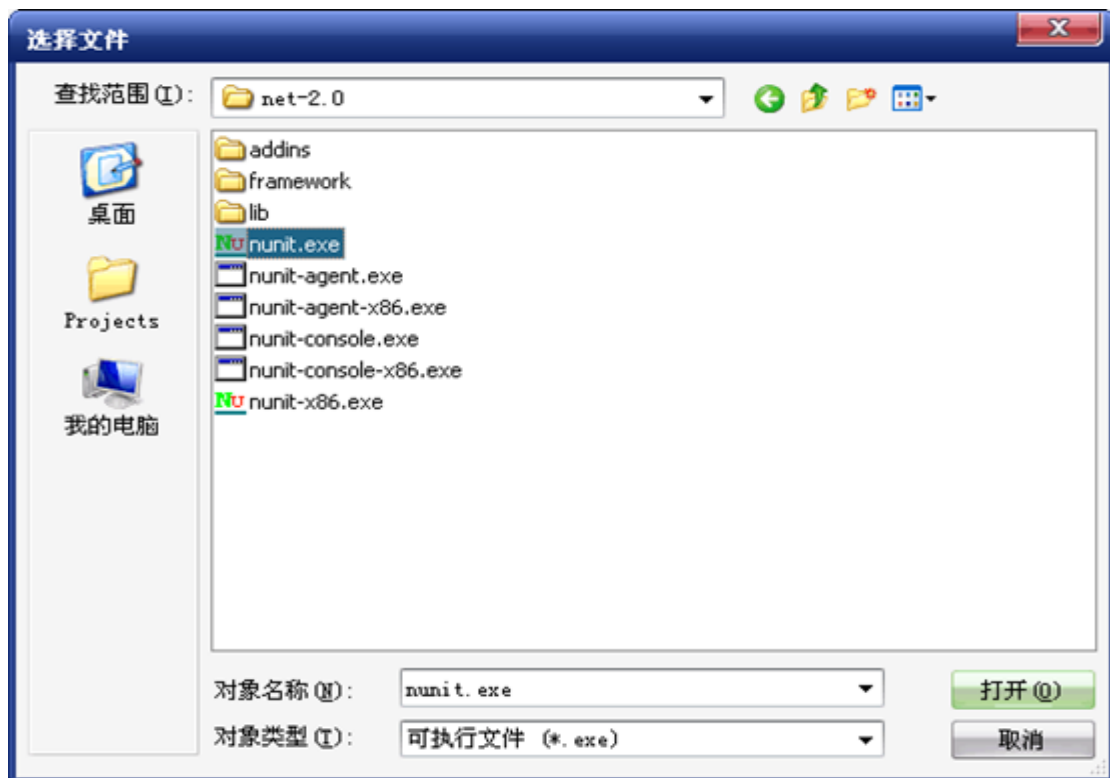


图 3-41 设置 NUnit 为调试程序 4

(5) 编译运行测试

生成解决方案成功后，点击启动调试，NUnit 测试运行器出现。也可以直接打开 NUnit 程序，从 File 菜单选择 Open Project，查找并选择 NUnitQuickStart.dll 的路径，加载本工程编译的程序集。当加载了测试的程序集，测试运行器为加载的程序集的测试产生一个可见的表现。

按 Run 按钮，树的节点变成绿色，而且测试运行器窗口上的进度条变绿，代表成功通过测试。反之如果有一个测试用例不成功，则为红色。

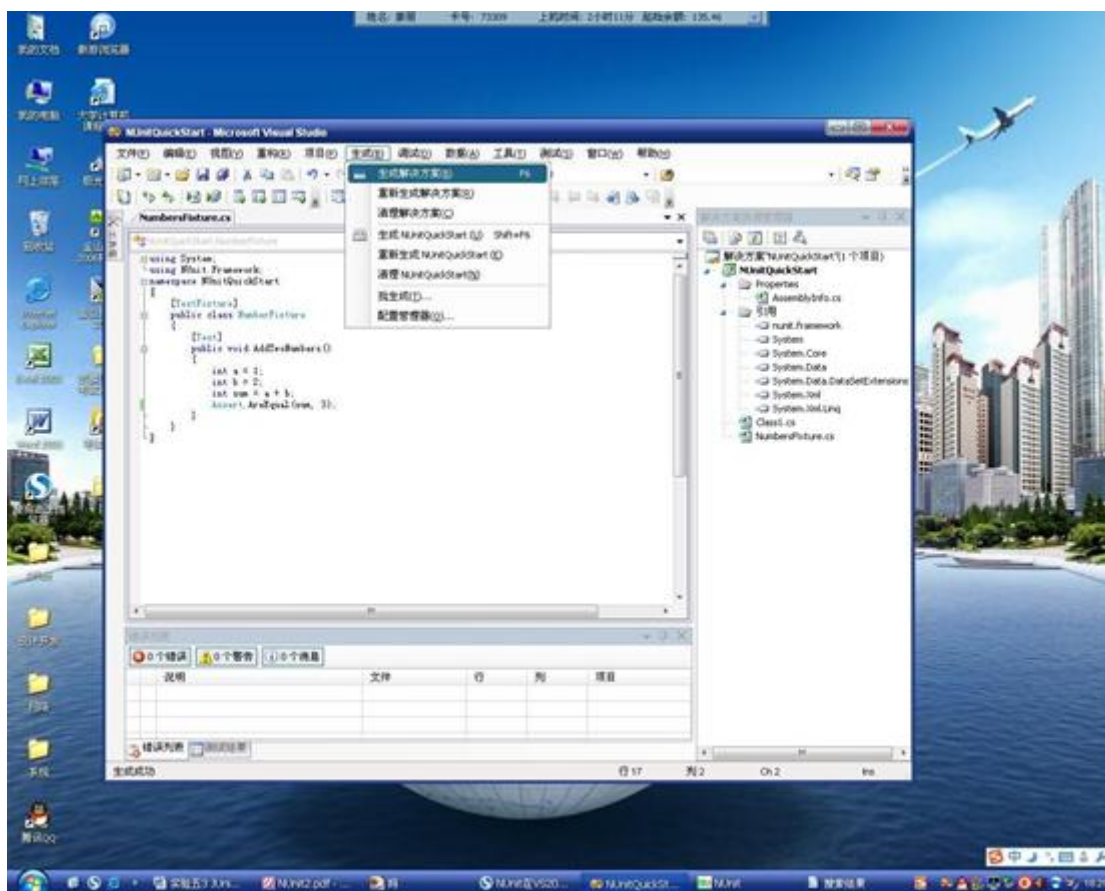


图 3-42 生成解决方案

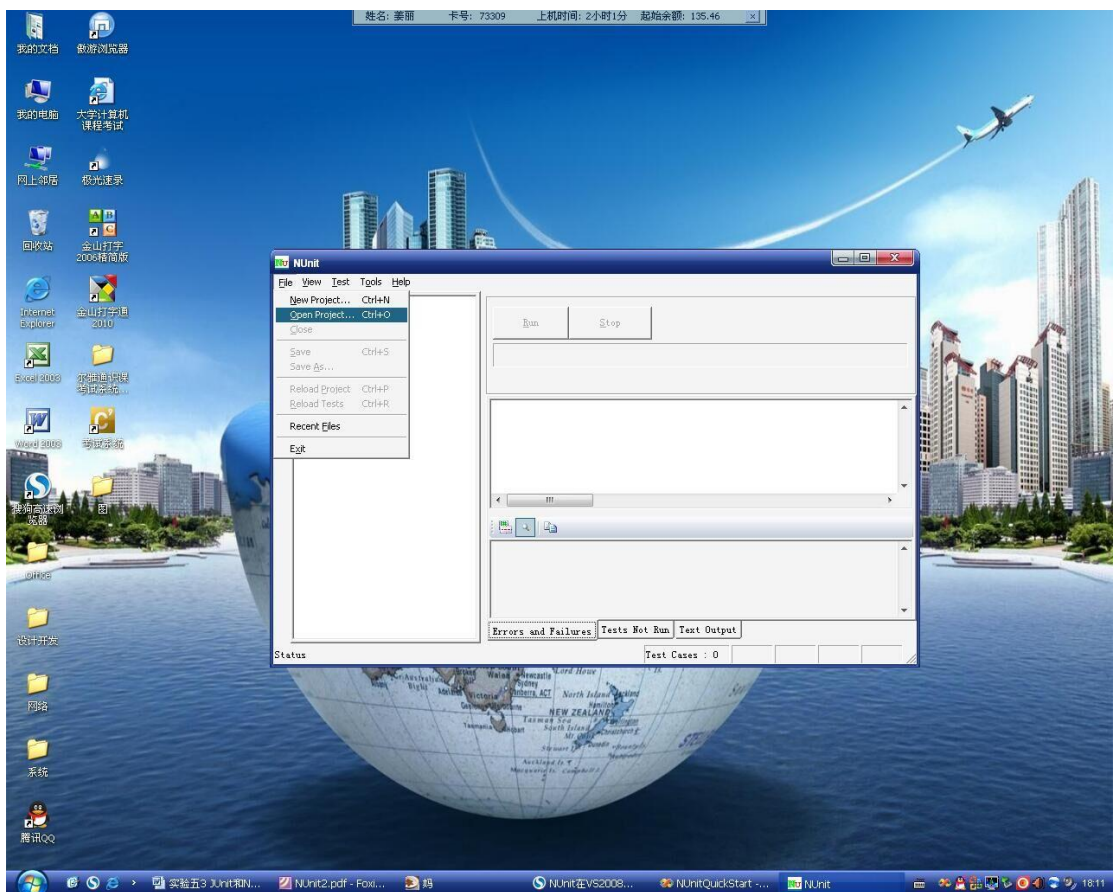


图 3-43 添加测试的程序集 1

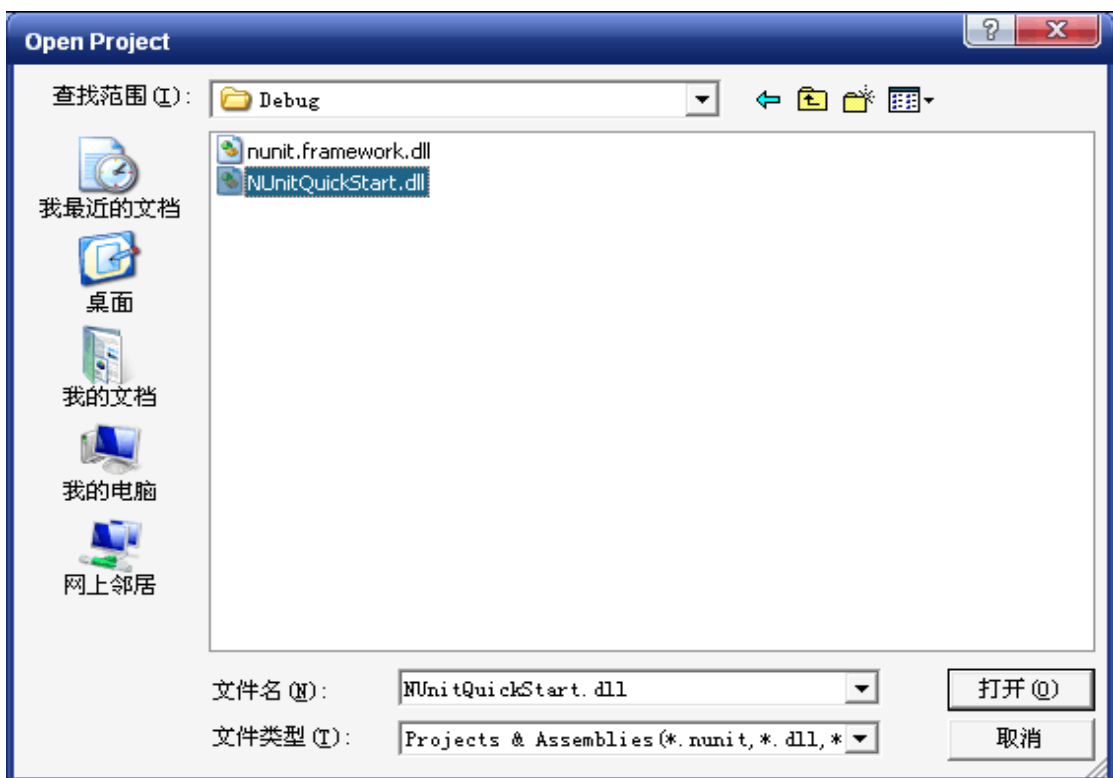


图 3-44 添加测试的程序集 2

添加测试的程序集成功后，显示出程序集的树型结构：

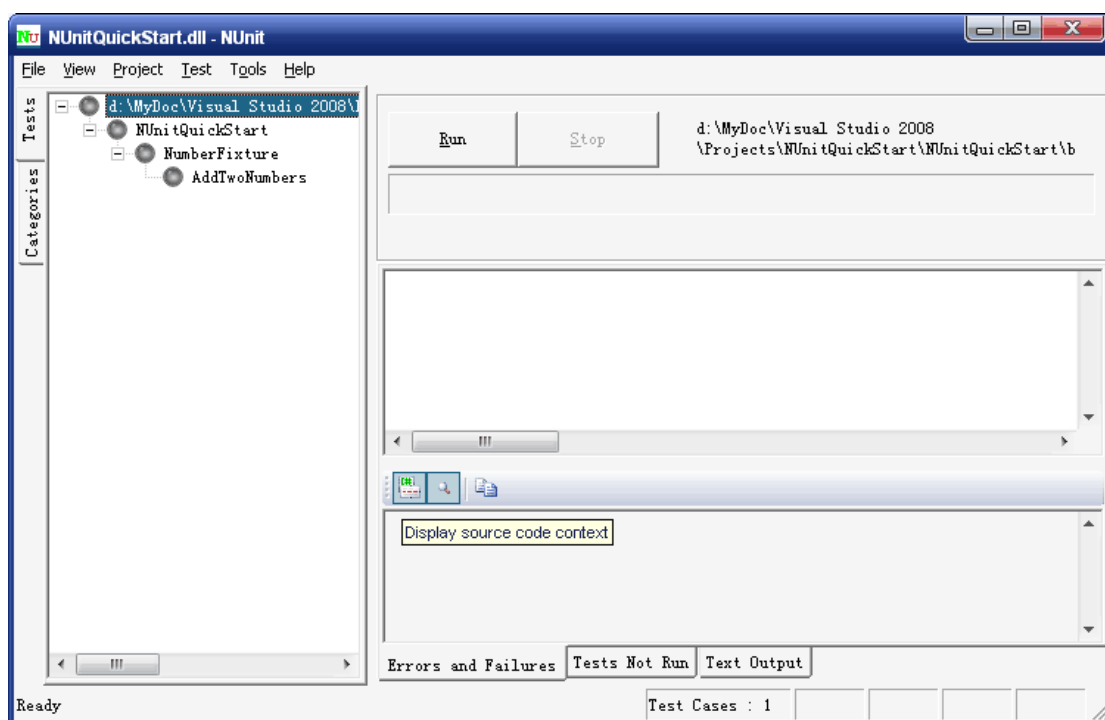


图 3-45 成功添加测试的程序集

点击 Run，树型结构显示绿色，进度条也变绿，代表成功通过测试。

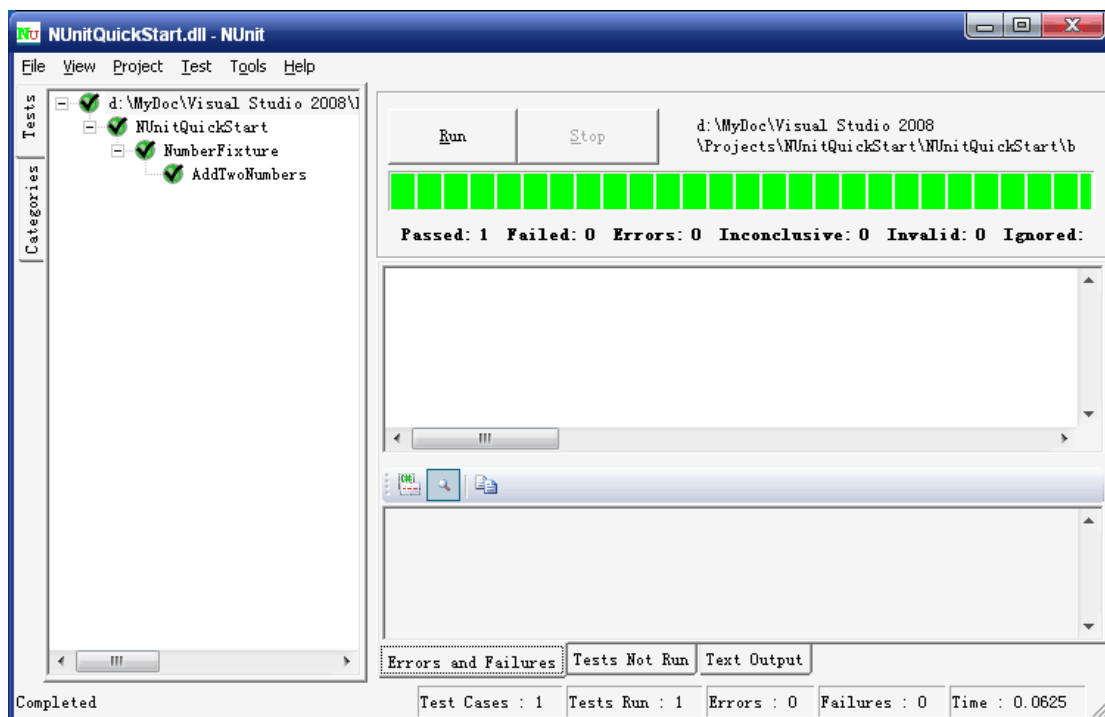


图 3-46 通过测试

```
using System;  
using NUnit.Framework;
```

```
namespace NUnitQuickStart
{
    [TestFixture]
    public class NumberFixture
    {
        [Test]
        public void AddTwoNumbers()
        {
            int a = 1;
            int b = 2;
            int sum = a + b;
            Assert.AreEqual(sum, 4);
        }
    }
}
```

修改一下测试用例，将 3 改为 4，这里只是说明效果，并不是正确的测试用例。

生成解决方案后，重复上述（5）步骤，得到下面的界面，树型结构显示红色，进度条也变红，代表测试出错。

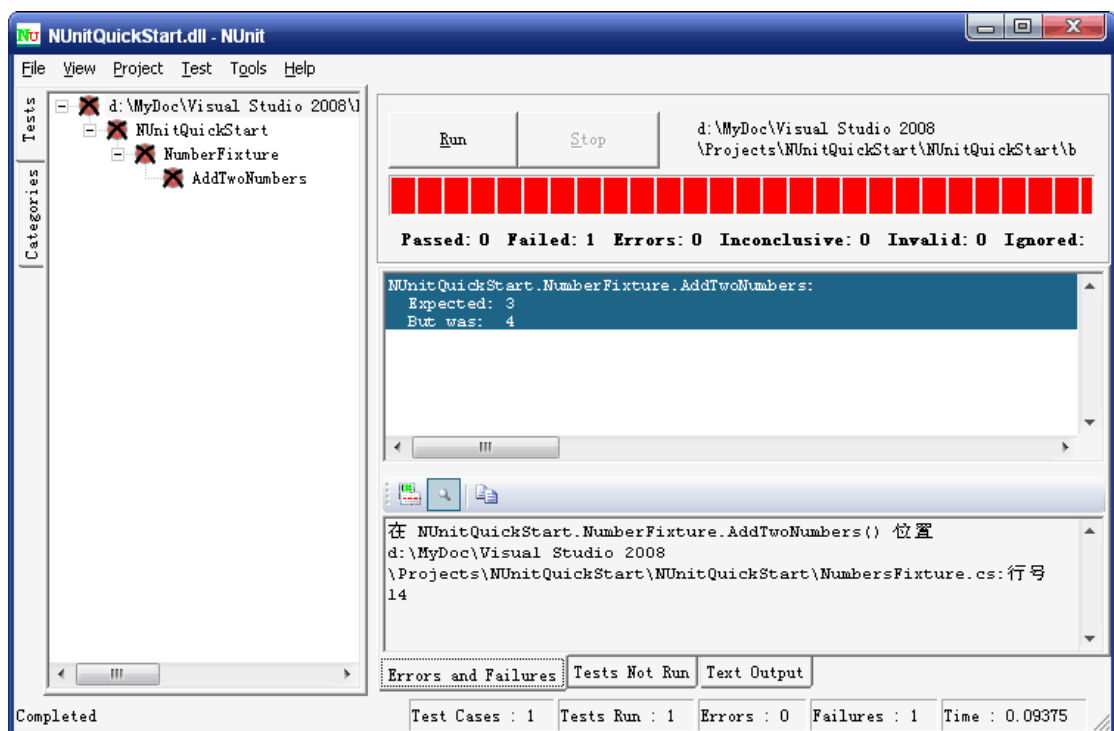


图 3-47 测试出错

4 实验体会

测试项目是整个软件开发过程中工作量最大的一项工作，必须高度重视软件测试工作。通过本次实验，我学会了 JUnit 插件对 Java 集成开发工具 Eclipse 进行白盒测试用例设计及自动测试，和 NUnit 程序对 .Net 集成开发工具 Visual Studio2008 进行白盒测试用例设计及自动测试，掌握了 Junit 和 NUnit 的使用方法。