专业化的单元测试工具

**一.写一个测试MathUtil的add方法的测试方法**

给一个方法加上@Test注解,那么该方法就是一个测试方法(就是一个main方法)

Junit测试方法的规则:

1)测试方法一定要在方法顶部加上@Test注解

2)测试的方法必须是public修饰,返回类型必须是void,不能有参数,可以抛出异常

**二.如何运行测试方法:**

1)测试某一个测试方法:光标直接移动到函数名的那一行上,直接alt + enter,选择第一个

2)测试所有的测试方法:光标直接移动到测试类的类名称的那一行上,直接alt + enter,选择第一个

3)运行一个项目的所有测试方法:直接右击项目名称->Run 'All Tests'

MathUtil.java

|  |
| --- |
| package util;  public class MathUtil {  //加法  public int add(int a,int b){  return a + b;  }  //除法  public int div(int a,int b){  return a/b;  }  } |

MathUtilTest.java(测试类)

|  |
| --- |
| package util;  import org.junit.Assert;  import org.junit.Test;  public class MathUtilTest{  @Test  public void addTest(){  MathUtil mathUtil = new MathUtil();  int result = mathUtil.add(10,5);  /\*\*  \* 手动判断结果,这种方式判断是不专业的  \*/  // if(result!=15){  // throw new RuntimeException("结果出错");  // }  /\*\*  \* 使用Junit专业的判定结果的方法:  \* Assert类:断言类(判断结果是什么?)  \*/  //左边是期望值,右边是要判定的值  Assert.assertEquals(15,result);//判断两个值是否相等,equal  // Assert.assertNotSame(15,result);//判断两个值是否不相等即!=  Assert.assertSame(15,result);//判断对象的地址是否相等即==  }  @Test  public void testDiv(){  MathUtil mathUtil = new MathUtil();  int result = mathUtil.div(10,5);  if(result!=2){  throw new RuntimeException("结果出错");  }  }  } |

快速生成测试类的方式在idea操作文档中有