### 辽师

### 《软件工程报告》

——JUNIT测试实验报告

### 

学 院：计算机与信息技术学院

专 业：计算机科学与技术（师范）

班级序号：01班

学 号：201621012299

学生姓名：周惠馨

完成时间：2018.11.23

1. 旧断言的测试

1、T1的单独测试x+y

T1.java:

**package** T;

**public** **class** T1 {

**public** **int** add(**int** x,**int** y){

**return** x+y;

}

}

T1Test.java:

**package** Test;

**import** org.junit.Test;

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

**import** T.T1;

**public** **class** T1Test {

@Test

**public** **void** testAdd() {

T1 a=**new** T1();

**int** s=a.add(3, 5);

*assertEquals*(8,a.add(3,5));

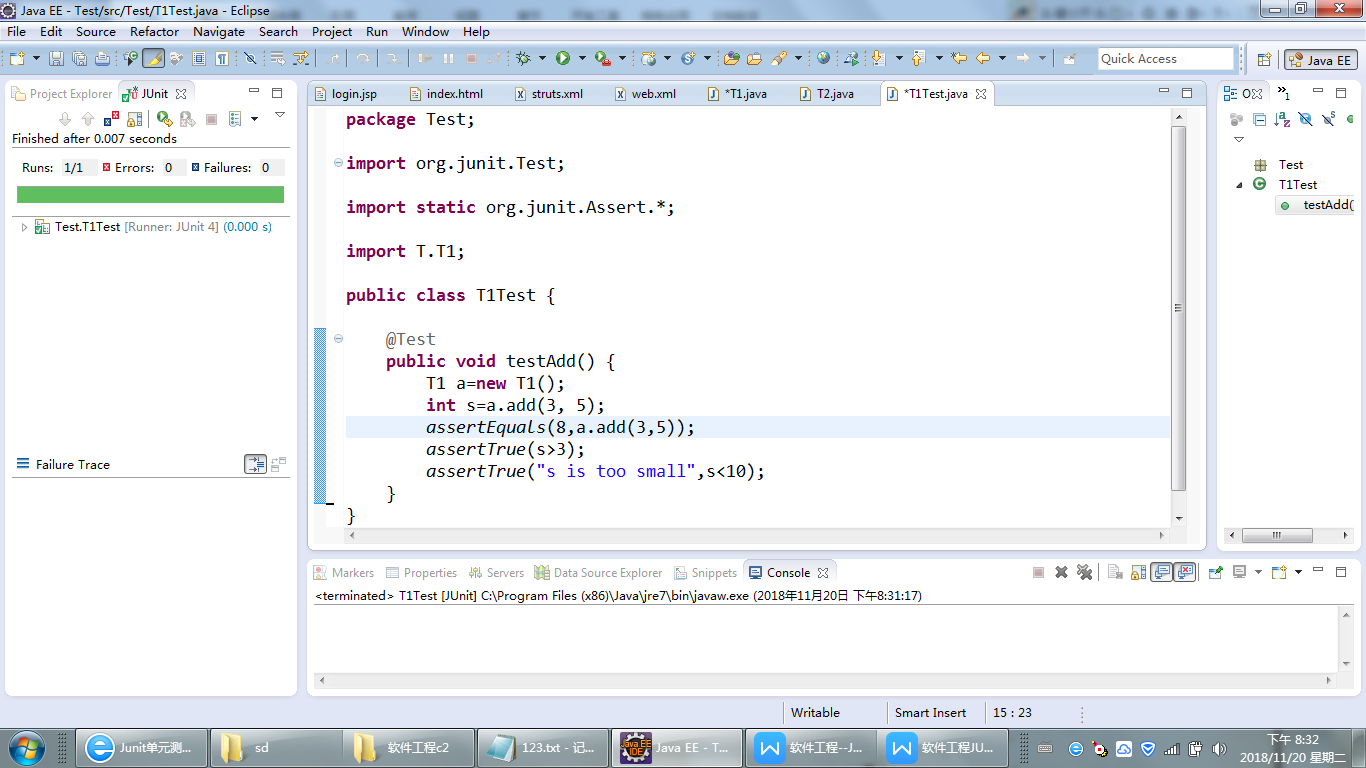
*assertTrue*(s>3);

*assertTrue*("s is too small",s<10);

}

}

正确测试实验结果（截图）



1. T1和T2的多个类多方法测试

T1.java、T1Test.Java代码同上

T2.java：

**package** T;

**public** **class** T2 {

**public** **int** divide(**int** x,**int** y){

**return** x/y;

}

**public** String getName(){

**return** ("夜莺");

}

}

T2Test.java：

**package** Test;

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

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

**import** org.junit.Test;

**import** T.T2;

**public** **class** T2Test {

@Test

**public** **void** testDivide() {

T2 b=**new** T2();

**int** s=b.divide(8,4);

*assertEquals*(2,b.divide(8,4));

*assertTrue*(s<5);

//assertTrue("s is too small",s>10);

}

@Test

**public** **void** testGetName() {

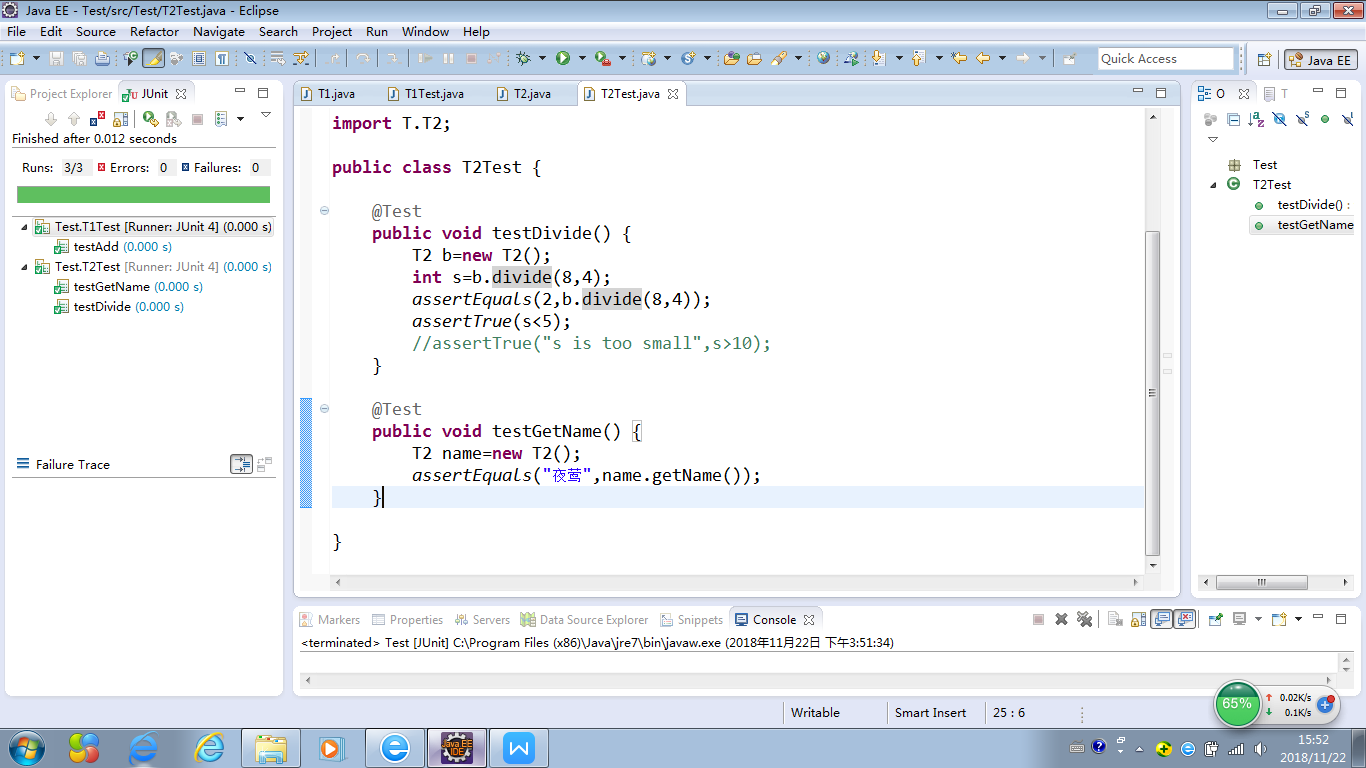
T2 name=**new** T2();

*assertEquals*("夜莺",name.getName());

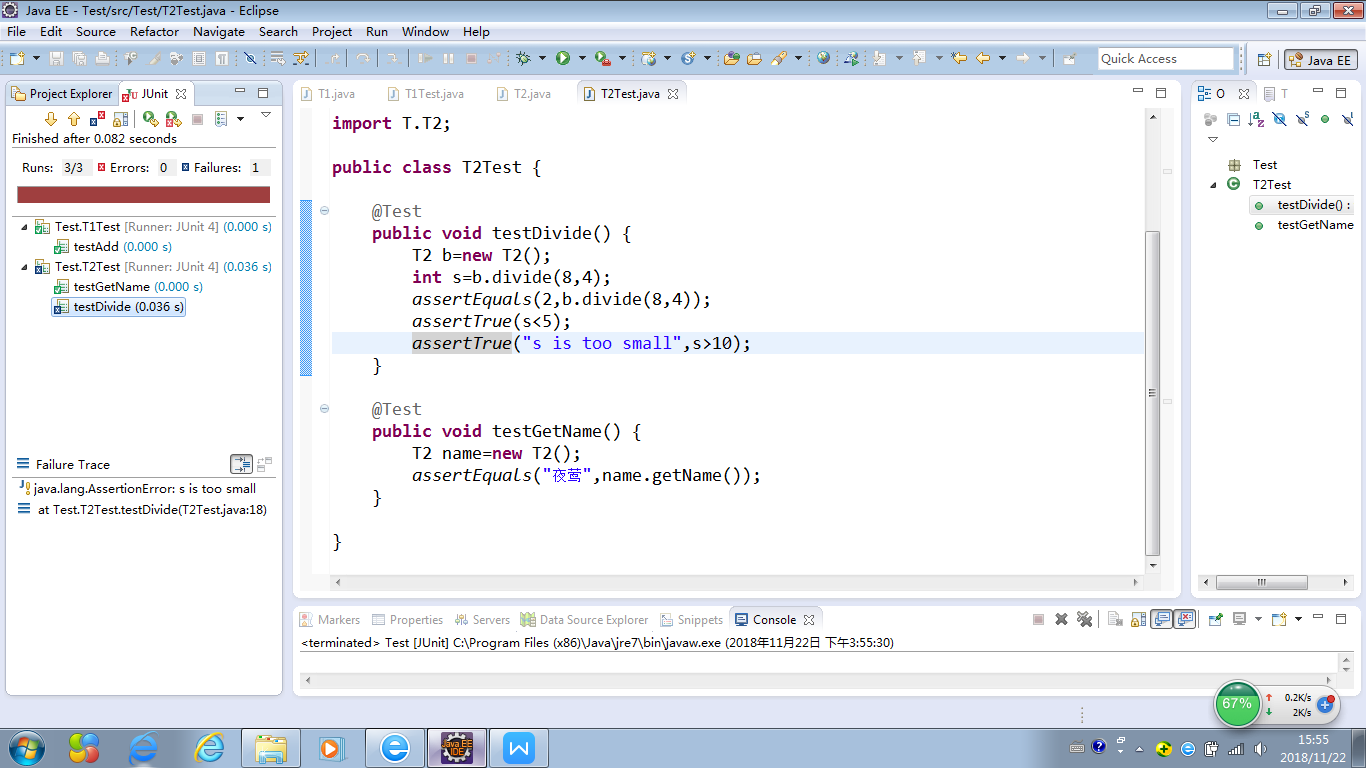
}

}

正确测试实验结果（截图）



错误测试实验结果（截图）



1. 新断言的测试
2. 新断言hamcrest测试
3. 测试1

T3.java:

**package** T;

**public** **class** T3 {

**public** **int** add(**int** x,**int** y){

**return** x+y;

}

}

T3Test.java:

**package** Test;

**import** **static** org.hamcrest.Matchers.*allOf*;

**import** **static** org.hamcrest.Matchers.*anyOf*;

**import** **static** org.hamcrest.Matchers.*anything*;

**import** **static** org.hamcrest.Matchers.*greaterThan*;

**import** **static** org.hamcrest.Matchers.~~is~~;

**import** **static** org.hamcrest.Matchers.*lessThan*;

**import** **static** org.hamcrest.Matchers.*not*;

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

**import** org.junit.Test;

**import** T.T3;

**public** **class** T3Test {

@Test

**public** **void** testAdd() {

**int** n=**new** T3().add(4,8);

//allOf匹配符表明如果接下来的所有条件必须都成立测试才通过

*assertThat*(n,*allOf*(*greaterThan*(1),*lessThan*(17)));

//anyOf匹配符表明如果接下来的所有条件只要有一个成立则测试通过

*assertThat*(n,*anyOf*(*greaterThan*(7),*lessThan*(7)));

//anything匹配符表明无论什么条件，永远为true

*assertThat*(n,*anything*());

//is匹配符表明如果前面待测的object等于后面给出的object，则测试通过

*assertThat*(n,*is*(12));

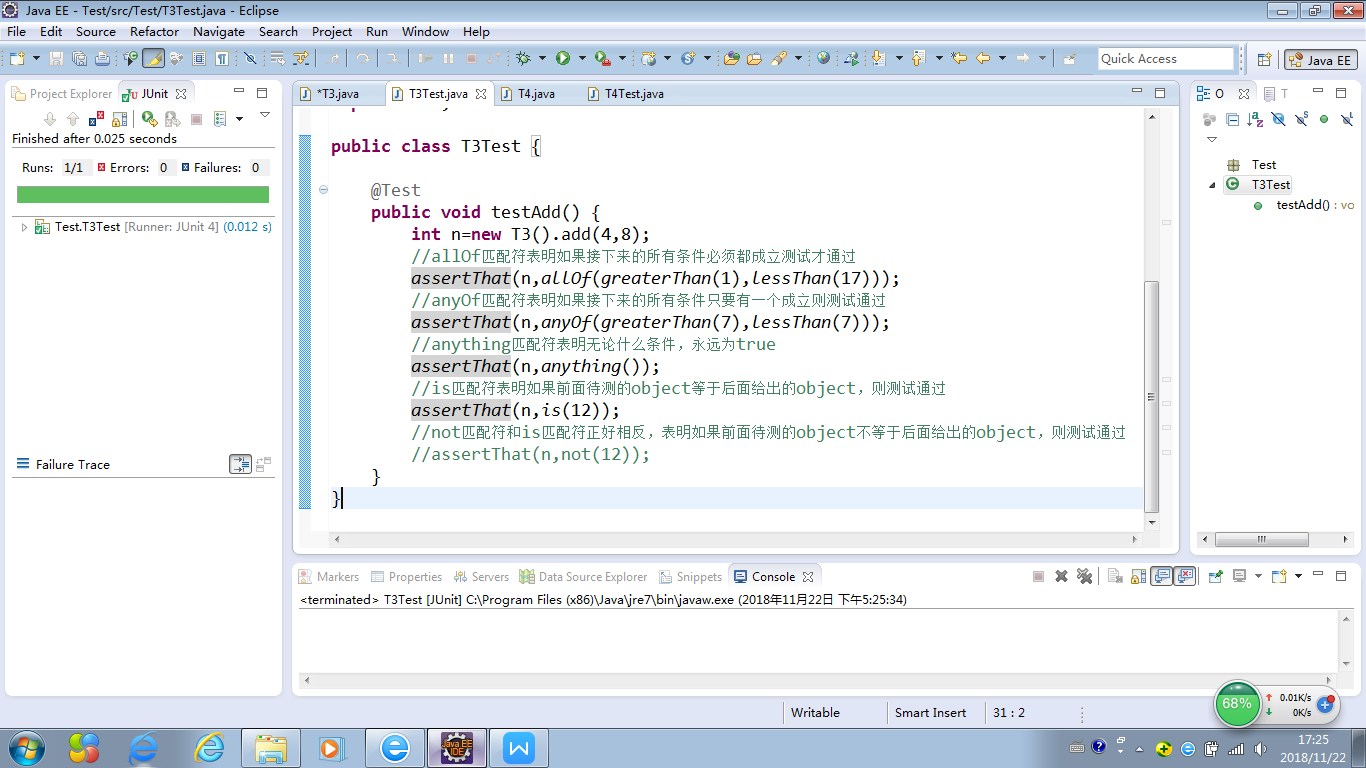
//not匹配符和is匹配符正好相反，表明如果前面待测的object不等于后面给出的object，则测试通过

//assertThat(n,not(12));

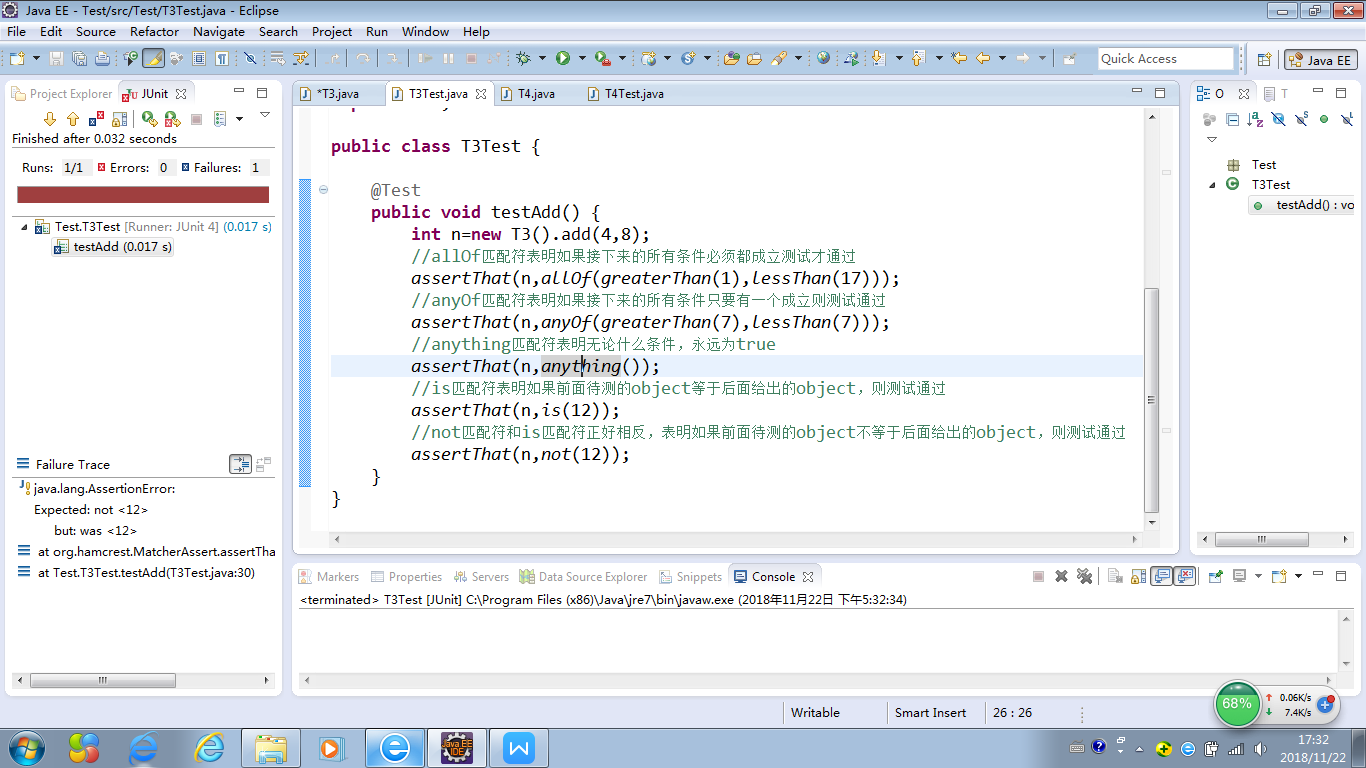
}

}

正确测试实验结果（截图）



错误测试实验结果（截图）



1. 测试2

T4.java:

**package** T;

**public** **class** T4 {

**public** String getName(){

**return** "testsuccess";

}

}

T4Test.java:

**package** Test;

**import** **static** org.hamcrest.Matchers.*containsString*;

**import** **static** org.hamcrest.Matchers.*endsWith*;

**import** **static** org.hamcrest.Matchers.*equalTo*;

**import** **static** org.hamcrest.Matchers.*equalToIgnoringCase*;

**import** **static** org.hamcrest.Matchers.*startsWith*;

**import** **static** org.hamcrest.Matchers.*equalToIgnoringWhiteSpace*;

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

**import** org.junit.Test;

**import** T.T4;

**public** **class** T4Test {

@Test

**public** **void** testGetName() {

String str=**new** T4().getName();

String str2="testsuccess";

//containsString匹配符表明如果测试的字符串str包含子字符串"test"则测试通过

*assertThat*(str,*containsString*("test"));

//endsWith匹配符表明如果测试的字符串str以子字符串"ss"结尾则测试通过

*assertThat*(str,*endsWith*("ss"));

//startsWith匹配符表明如果测试的字符串str以子字符串"testsucc"开始则测试通过

*assertThat*(str,*startsWith*("testsucc"));

// equalTo匹配符表明如果测试的str等于str2则测试通过

*assertThat*(str,*equalTo*(str2));

//equalToIgnoringCase匹配符表明如果测试的字符串str在忽略大小写的情况下等于"TestSuccess"则测试通过

*assertThat*(str,*equalToIgnoringCase*("TestSuccess"));

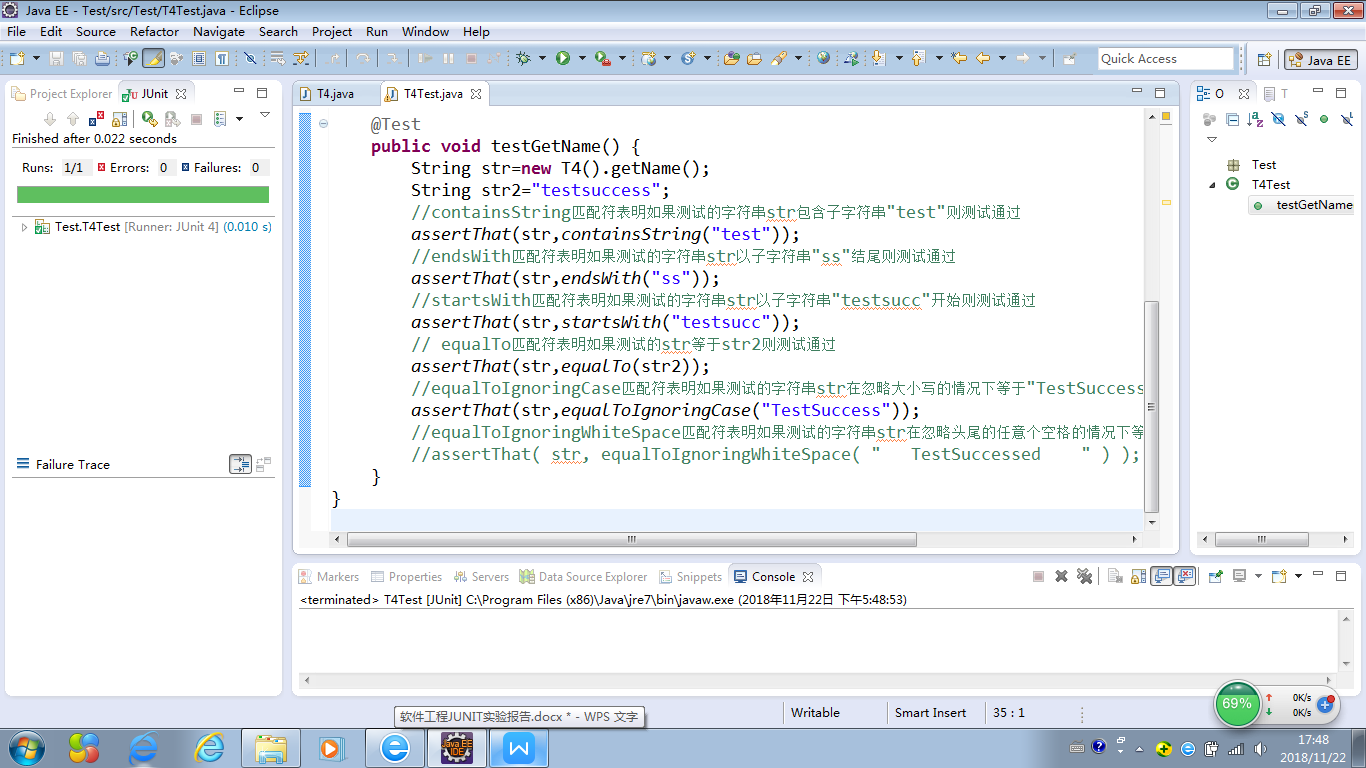
//equalToIgnoringWhiteSpace匹配符表明如果测试的字符串str在忽略头尾的任意个空格的情况下等于" testsuccess "则测试通过

//assertThat( str, equalToIgnoringWhiteSpace( " TestSuccessed " ) );

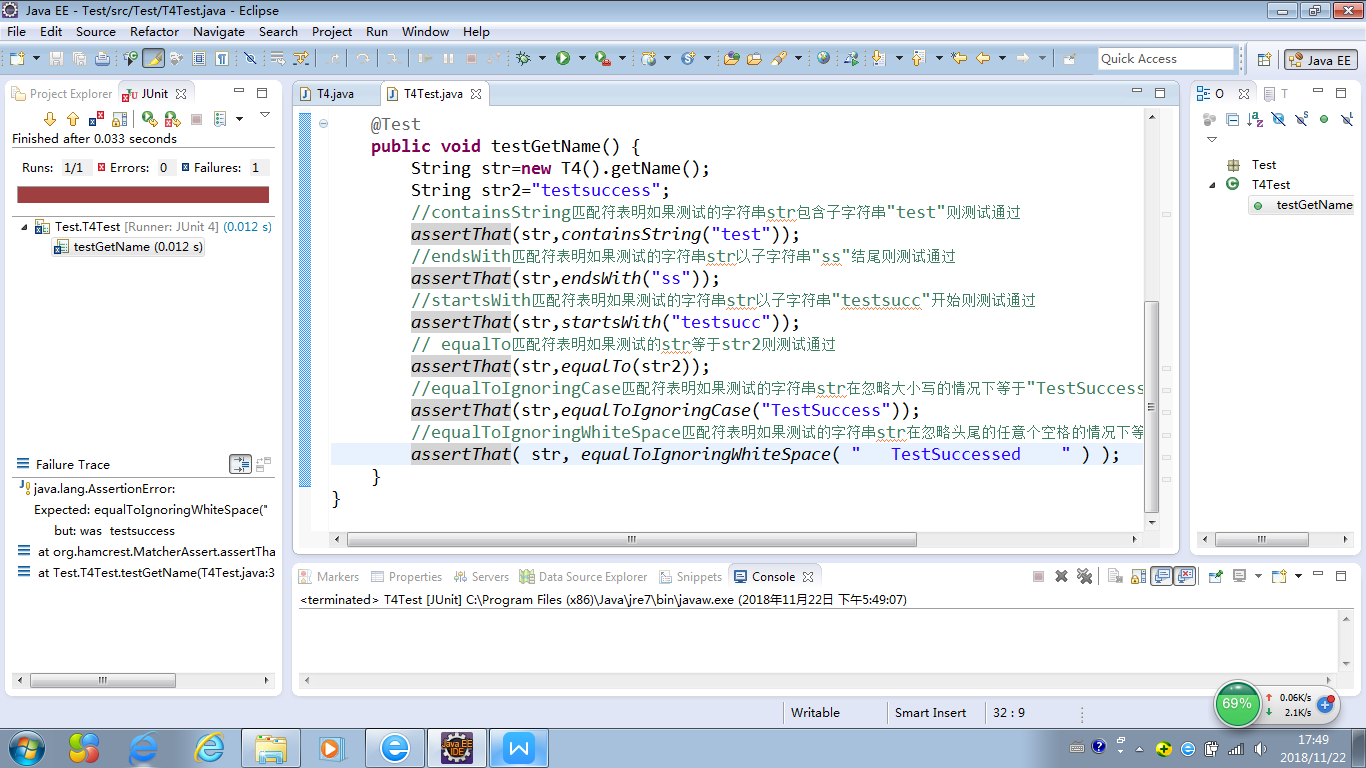
}

}

正确测试实验结果（截图）



错误测试实验结果（截图）



1. 测试3

T5.java:

**package** T;

**public** **class** T5 {

**public** **double** mult(**double** x,**double** y){

**return** x\*y;

}

}

T5Test.java:

**package** Test;

**import** **static** org.hamcrest.Matchers.*closeTo*;

**import** **static** org.hamcrest.Matchers.*greaterThan*;

**import** **static** org.hamcrest.Matchers.*greaterThanOrEqualTo*;

**import** **static** org.hamcrest.Matchers.*lessThan*;

**import** **static** org.hamcrest.Matchers.*lessThanOrEqualTo*;

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

**import** org.junit.Test;

**import** T.T5;

**public** **class** T5Test {

@Test

**public** **void** testMult() {

**double** c=**new** T5().mult(4,7);

//closeTo匹配符表明如果所测试的浮点型数c在20.0±0.5范围之内则测试通过

*assertThat*(c,*closeTo*(24.0,4.0));

//greaterThan匹配符表明如果所测试的数值c大于16.0则测试通过

*assertThat*(c,*greaterThan*(20.0));

//lessThan匹配符表明如果所测试的数值c小于16.0则测试通过

*assertThat*(c,*lessThan*(30.0));

//greaterThanOrEqualTo匹配符表明如果所测试的数值c大于等于16.0则测试通过

*assertThat*(c,*greaterThanOrEqualTo*(28.0));

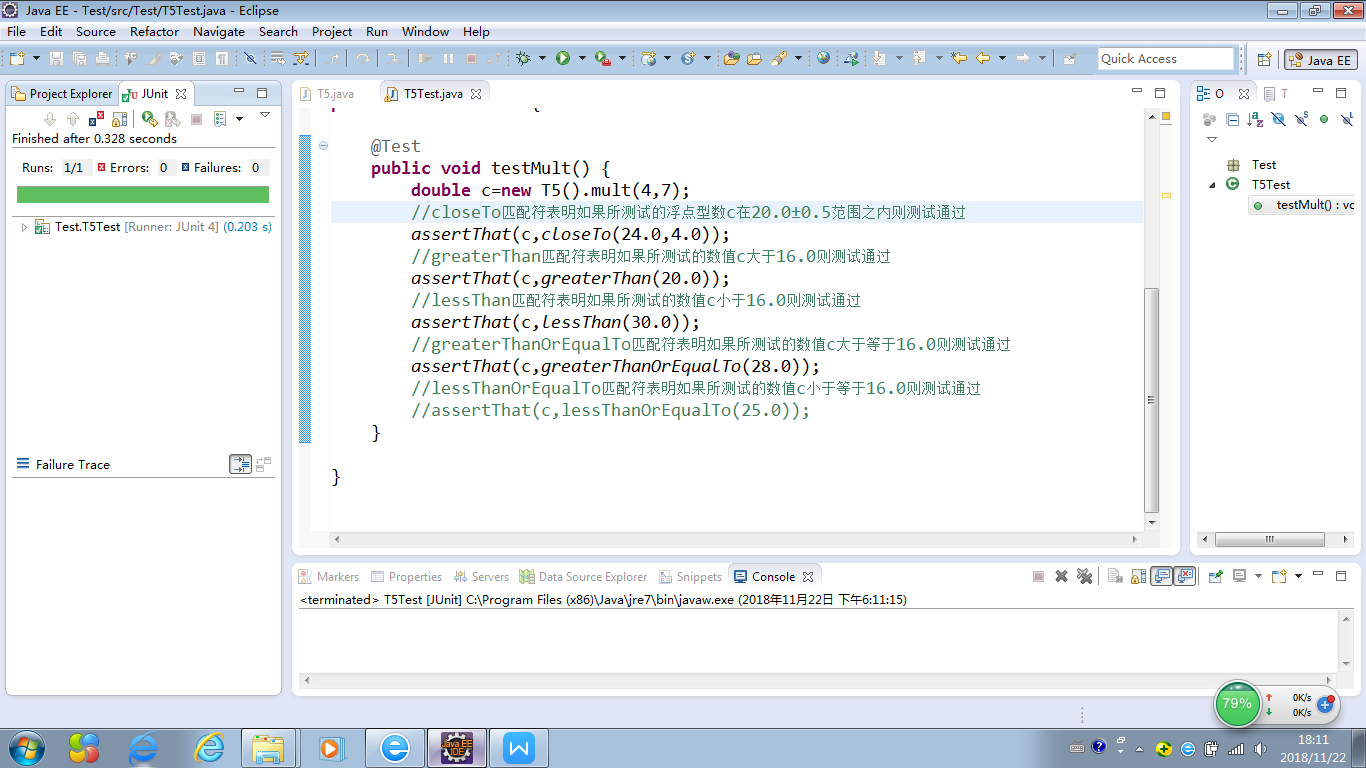
//lessThanOrEqualTo匹配符表明如果所测试的数值c小于等于16.0则测试通过

//assertThat(c,lessThanOrEqualTo(25.0));

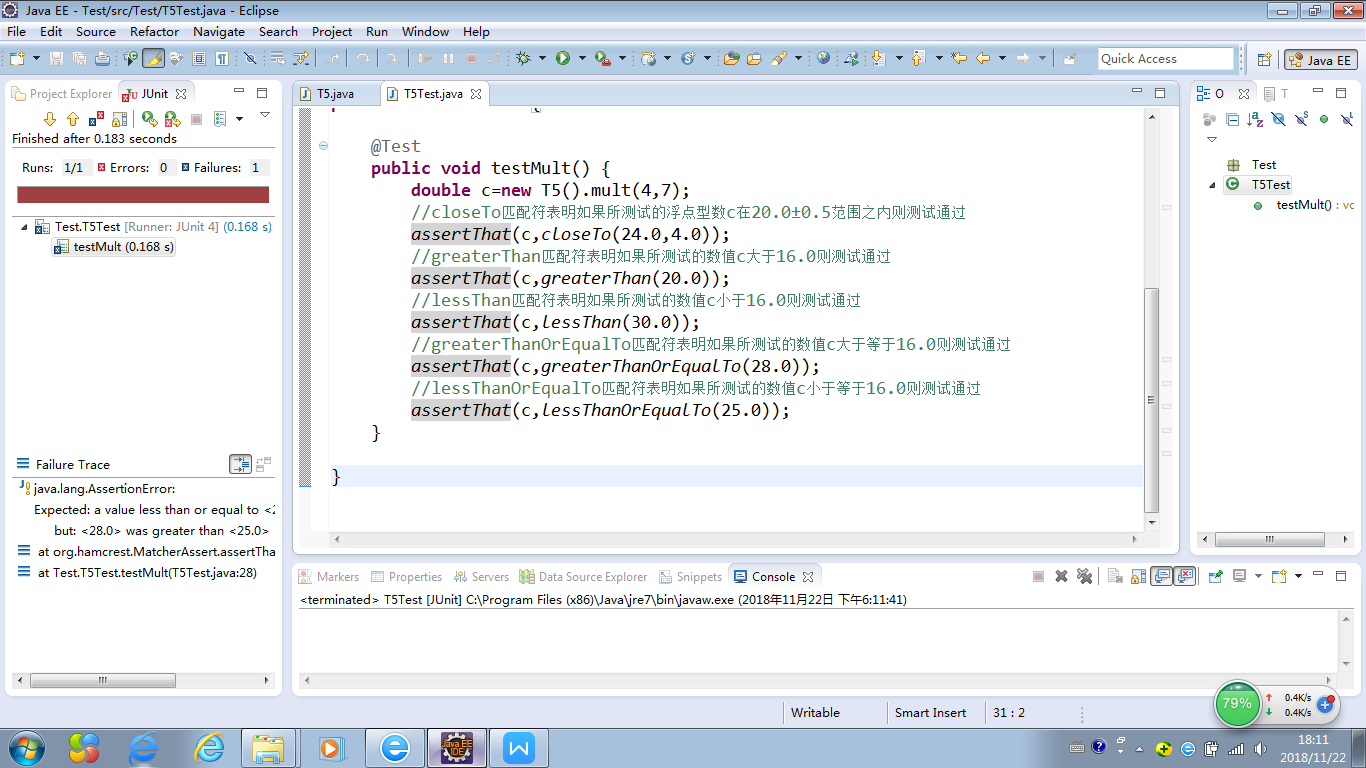
}

}

正确测试实验结果（截图）



错误测试实验结果（截图）



1. Annocation的运用

T6.java:

**package** T;

**public** **class** T6 {

**public** **int** sub(**int** x,**int** y){

**return** x-y;

}

}

T6Test.java:

**package** Test;

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

**import** **static** org.hamcrest.Matchers.\*;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**import** T.T6;

**public** **class** T6Test {

@Before

**public** **void** before() {

System.*out*.println("before");

}

@BeforeClass

**public** **static** **void** beforeClass() {

System.*out*.println("beforeClass");

}

@Test

**public** **void** testSub() {

**int** d=**new** T6().sub(10, 6);

*assertThat*(d,*is*(4));

*assertThat*(d,*allOf*(*greaterThan*(3),*lessThan*(5)));

}

@After

**public** **void** After() {

System.*out*.println("after");

}

@AfterClass

**public** **static** **void** AfterClass() {

System.*out*.println("afterClass");

}

}

测试实验结果（截图）

