**软件质量保证与测试**

**实习报告**



**班 级：\_ \_20软工A2\_\_\_\_**

**姓 名：** \_\_ 陆恺琦\_\_\_\_\_

**学 号：\_\_\_20201111339\_\_\_**

**序 号： \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_**

**日 期： 2023-06-11\_ \_**

# 1 Triangle

## 试题描述

三角形Triangle，测试输入的3条边是否构成三角形，是什么类型的三角形。测试数据如：（2，3，4），（6，6，6），（5，5，6），（1，1，5），（-1，1，5）等。

## 流程图



## 测试代码

import org.junit.Test;

import static org.junit.Assert.\*;

public class TriangleTest {

Triangle T1 = new Triangle(2,3,4);

Triangle T2 = new Triangle(6,6,6);

Triangle T3 = new Triangle(5,5,6);

Triangle T4 = new Triangle(1,1,5);

Triangle T5 = new Triangle(-1,1,5);

@Test

public void testIsTriangle(){

assertEquals(true,T1.isTriangle(T1));

assertEquals(true,T2.isTriangle(T2));

assertEquals(true,T3.isTriangle(T3));

assertEquals(false,T4.isTriangle(T4));

assertEquals(false,T5.isTriangle(T5));

}

@Test

public void testGetType(){

assertEquals("Scalene",T1.getType(T1));

assertEquals("Regular",T2.getType(T2));

assertEquals("Isosceles",T3.getType(T3));

assertEquals("Illegal",T4.getType(T4));

assertEquals("Illegal",T5.getType(T5));

}

@Test

public void testDiffOfBorders(){

assertEquals(1,T1.diffOfBorders(2,3));

}

@Test

public void testGetBorders(){

long[] borders = new long[3];

borders[0]=2;

borders[1]=3;

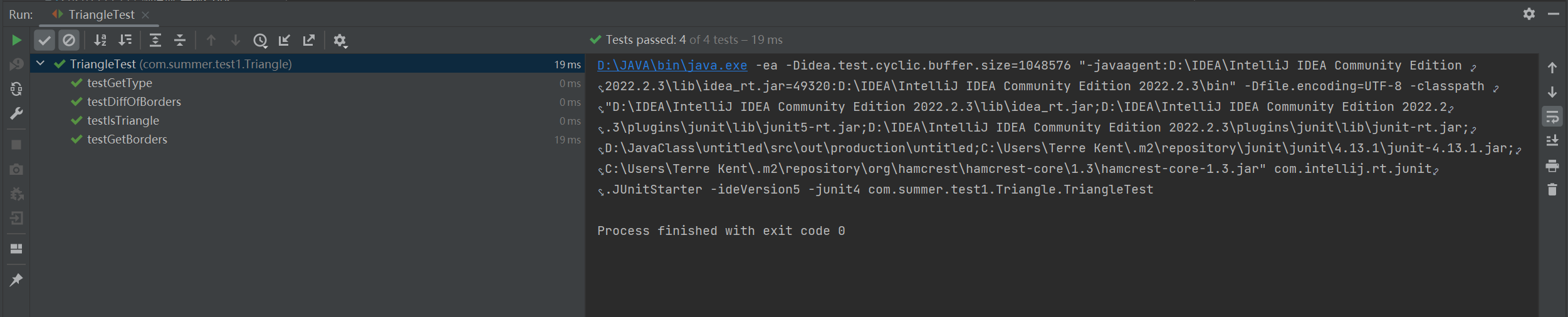
borders[2]=4;

assertArrayEquals(borders,T1.getBorders());

}

}

## Idea测试截图



# 2 Nextday

## 试题描述

NextDay是一个简单的日期计算器，计算给定日期的下一天的具体日期。如给定2020年1月1日，返回2020年1月2日；给定2020年1月31日，返回2020年2月1日。

## 流程图



测试用例如下：



## 测试代码

import org.junit.Assert;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.junit.matchers.JUnitMatchers.containsString;

public class NextdayTest {

@Test

public void test(){

Date date=new Date(6,1,2023);

Date d = Nextday.nextDay(date);

String s= d.toString();

Assert.assertEquals("6/2/2023",s);

}

@Test

public void test01(){

Date date=new Date(2,28,2023);

Date d = Nextday.nextDay(date);

String s= d.toString();

Assert.assertEquals("3/1/2023",s);

}

@Test

public void test02(){

Date date=new Date(2,29,2024);

Date d = Nextday.nextDay(date);

String s= d.toString();

Assert.assertEquals("3/1/2024",s);

}

@Test

public void test03(){

Date date=new Date(1,31,2023);

Date d = Nextday.nextDay(date);

String s= d.toString();

Assert.assertEquals("2/1/2023",s);

}

@Test

public void test04(){

Date date=new Date(6,30,2023);

Date d = Nextday.nextDay(date);

String s= d.toString();

Assert.assertEquals("7/1/2023",s);

}

@Test

public void test05(){

Date date=new Date(12,31,2023);

Date d = Nextday.nextDay(date);

String s= d.toString();

Assert.assertEquals("1/1/2024",s);

}

@Test

public void testDate01(){

try{

Date date=new Date(-1,1,2023);

Nextday n=new Nextday();

Date d=Nextday.nextDay(date);

}catch(IllegalArgumentException e){

assertThat(e.getMessage(),containsString("Not a valid month"));

}

}

@Test

public void testDate02(){

try{

Date date=new Date(13,1,2023);

Nextday n=new Nextday();

Date d=Nextday.nextDay(date);

}catch(IllegalArgumentException e){

assertThat(e.getMessage(),containsString("Not a valid month"));

}

}

@Test

public void testDate03(){

try{

Date date=new Date(1,-1,2023);

Nextday n=new Nextday();

Date d=Nextday.nextDay(date);

}catch(IllegalArgumentException e){

assertThat(e.getMessage(),containsString("Not a valid day"));

}

}

@Test

public void testDate04(){

try{

Date date=new Date(2,29,2023);

Nextday n=new Nextday();

Date d=Nextday.nextDay(date);

}catch(IllegalArgumentException e){

assertThat(e.getMessage(),containsString("Not a valid day"));

}

}

@Test

public void testDate05(){

try{

Date date=new Date(3,32,2023);

Nextday n=new Nextday();

Date d=Nextday.nextDay(date);

}catch(IllegalArgumentException e){

assertThat(e.getMessage(),containsString("Not a valid day"));

}

}

@Test

public void testDate06(){

try{

Date date=new Date(6,31,2023);

Nextday n=new Nextday();

Date d=Nextday.nextDay(date);

}catch(IllegalArgumentException e){

assertThat(e.getMessage(),containsString("Not a valid day"));

}

}

}

## Idea测试截图

