

1、实验要求

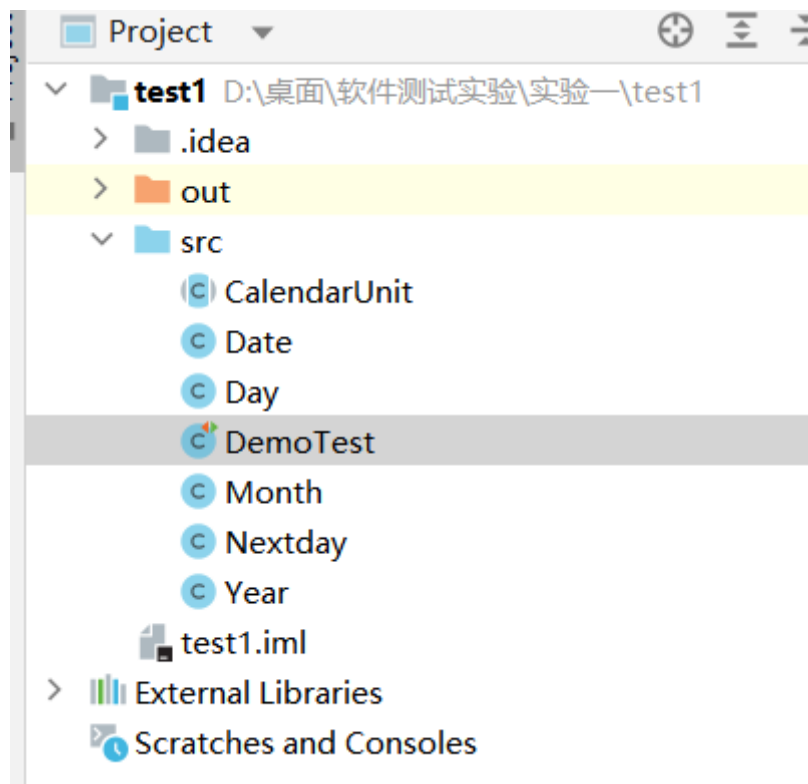
理解和掌握使用 Eclemma 软件测试工具对 Java 语言编写的程序进行语句覆盖测试。

根据提供的代码，编写程序并且进行覆盖率测试，要求覆盖率尽可能高。

2、实验过程

2.1 在idea中导入项目Nextday

如图所示



2.2 仿照示例代码编写测试代码

测试代码如下：

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

public class DemoTest {

    @Test
    public void test() {
        // 测试用例
        Date date1 = new Date(2, 29, 2020); // 日期1
        Date date2 = new Date(2, 28, 2022); // 日期2（用于和日期1比较）
        Date date3 = new Date(12, 31, 2022); // 日期3
        Date date4 = new Date(3, 1, 2022); // 日期4（用于和日期2的下一天比较）
        Date date5 = new Date(12, 31, -1); // 日期5

        // 计算日期的下一天
```

```

Date nd1 = Nextday.nextDay(date1);
Date nd2 = Nextday.nextDay(date2);
Date nd3 = Nextday.nextDay(date3);
Date nd4 = Nextday.nextDay(date4);
Date nd5 = Nextday.nextDay(date5);

// 转换日期对象为字符串
String result1 = nd1.toString();
String result2 = nd2.toString();
String result3 = nd3.toString();
String result4 = nd4.toString();
String result5 = nd5.toString();

// 测试Date类打印日期方法
nd1.printDate();
nd2.printDate();
nd3.printDate();
nd4.printDate();
nd5.printDate();

// 使用断言来验证预期的结果
// 测试Date类的toString方法
Assert.assertEquals("3/1/2020", result1);
Assert.assertEquals("3/1/2022", result2);
Assert.assertEquals("1/1/2023", result3);
Assert.assertEquals("3/2/2022", result4);
Assert.assertEquals("1/1/1", result5);

// 测试Date类的nextday方法
Assert.assertTrue(nd2.equals(date4));
// 测试Date类的equals方法
Assert.assertFalse(date1.equals(date2));
// 测试getDay、getYear、getMonth
Assert.assertFalse(date1.getYear().equals(1));
Assert.assertFalse(date1.getMonth().equals(1));
Assert.assertFalse(date1.getDay().equals(1));

// 测试异常处理
// 验证不合法日期参数会抛出异常，并且异常包含正确的错误消息

// 日期不规范（下溢）
try {
    Date date = new Date(12, -1, 2020);
} catch (IllegalArgumentException e) {
    Assert.assertEquals(e.getMessage(), "Not a valid day");
}
// 日期不规范（上溢）
try {
    Date date = new Date(12, 32, 2020);
} catch (IllegalArgumentException e) {
    Assert.assertEquals(e.getMessage(), "Not a valid day");
}
// 月份不规范（上溢）
try {
    Date date = new Date(13, 1, 2020);

```

```

    } catch (IllegalArgumentException e) {
        Assert.assertEquals(e.getMessage(), "Not a valid month");
    }
    // 月份不规范（下溢）
    try {
        Date date = new Date(-1, 1, 2020);
    } catch (IllegalArgumentException e) {
        Assert.assertEquals(e.getMessage(), "Not a valid month");
    }
    // 年份不规范（下溢）
    try {
        Date date = new Date(1, 1, 0);
    } catch (IllegalArgumentException e) {
        Assert.assertEquals(e.getMessage(), "Not a valid year");
    }
}
}

```

编程思路：

仔细查看源码，为每个功能，如nextday、equals等等，设计合适的测试用例，使所有功能点都能被覆盖。比如equals用assertFalse或者assertTrue就可以测试出来或者直接使用assertEquals方法。get方法可以直接调用。

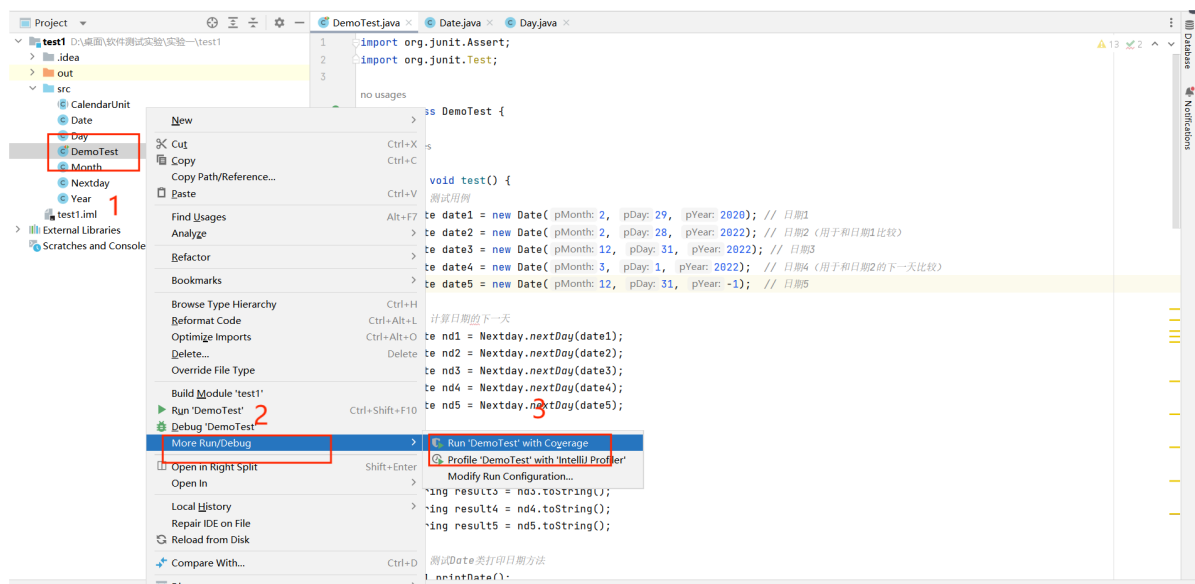
2.3 配置idea覆盖率测试环境

- 下载Apache Maven
- 找到maven配置项
- 修改maven配置
- 配置运行程序虚拟机参数
- 重启idea

参考教程：https://blog.csdn.net/m0_46491549/article/details/126578953?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522169851071116777224420835%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request_id=169851071116777224420835&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-2-126578953-null-null.142

2.4 进行覆盖率测试

程序运行方式：



覆盖率截图：

元素	类(%)	方法(%) ▲	行(%)
▼ 所有	100% (12...)	100% (62/62)	100% (184/184)
Nextday	100% (1/1)	100% (1/1)	100% (3/3)
CalendarUnit	100% (1/1)	100% (2/2)	100% (3/3)
Day	100% (1/1)	100% (6/6)	100% (20/20)
Year	100% (1/1)	100% (7/7)	100% (22/22)
Month	100% (1/1)	100% (7/7)	100% (25/25)
Date	100% (1/1)	100% (8/8)	100% (19/19)

测试表明，测试类能够覆盖Nextday中所有类、方法和语句。

3、实验分析

3.1 问题1：idea配置Mean

刚开始我试着运行示例代码的时候，发现idea运行不了，复制报错信息查找资料之后，发现是没有配置Mean，在跟着网络上的教程配置了Mean之后，运行成功。

3.2 问题2：测试用例的设计

在成功运行示例代码后，我尝试按照自己的思路写测试用例，但是发现覆盖率总是达不到100%。在仔细阅读了源码和大量尝试后，发现问题在于少测了get方法，修改后覆盖率达到100%

4、实验总结

这是第一次做软件测试的实验，刚开始做确实很懵，不知道如何下手，之后仔细看实验指导书，完成了本次实验。

本次实验比较简单，只要创建用例来测试每个方法即可。只要仔细查看源码，就可以创建出测试用例来进行所有的测试。比如equals用assertFalse或者assertTrue就可以测试出来或者直接使用assertEquals方法。get方法可以直接调用。

此外，因为我惯用idea，实验指导书上的示例使用的是eclipse，因此画了比较多的时间摸索idea进行覆盖率测试的方法，刚开始的时候一直在报错，后来参考了网上的教程配置MAVEN，成功运行。