**广州大学学生实验报告**

**开课学院及实验室：**AI实验室 **2023年11月2日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | **计算机科学与网络工程学院** | **年级/专业/班** | **软件211/212/213** | **姓名** |  | **学号** |  |
| **实验课程名称** | 软件质量保证与测试实验 | | | | | **成绩** |  |
| **实验项目名称** | 实验一 单元测试实验  实验二 功能测试实验 | | | | | **指导老师** | 李亚 |

教师评语：

**实验一 单元测试实验**

一、实验目的

1、掌握黑盒测试技术和白盒测试技术

2、熟练使用Junit测试框架进行基于Java语言的单元测试

3、掌握基于覆盖和基于独立路径的测试用例设计方法

4、掌握等价类划分法、边界值法、决策表法等测试用例设计方法

二、基本知识

1、单元测试概念

2、Junit测试框架的使用

3、黑盒/白盒测试技术

三、实验环境

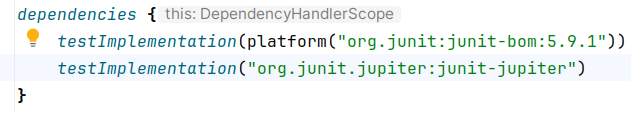
①NeoKylin操作系统+浏览器

②Jetbrains IntelliJ IDEA集成开发环境+Junit 5测试框架

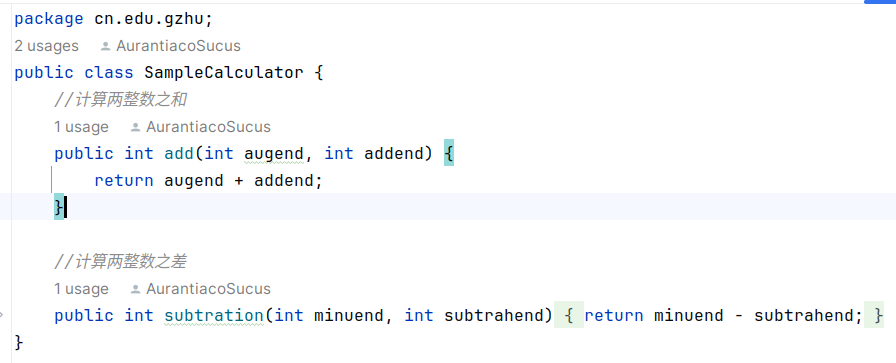
四、实验内容

1、学习Junit框架的使用

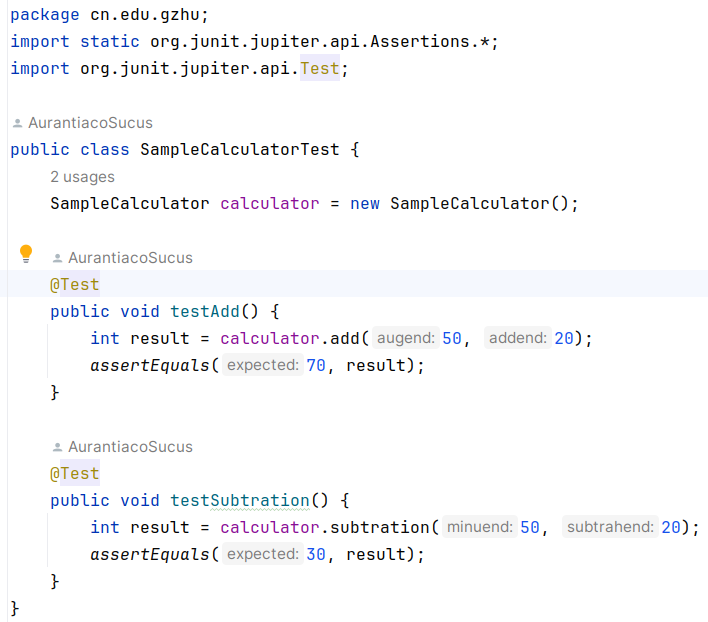
使用IDEA创建Gradle项目，添加JUnit Jupiter支持。



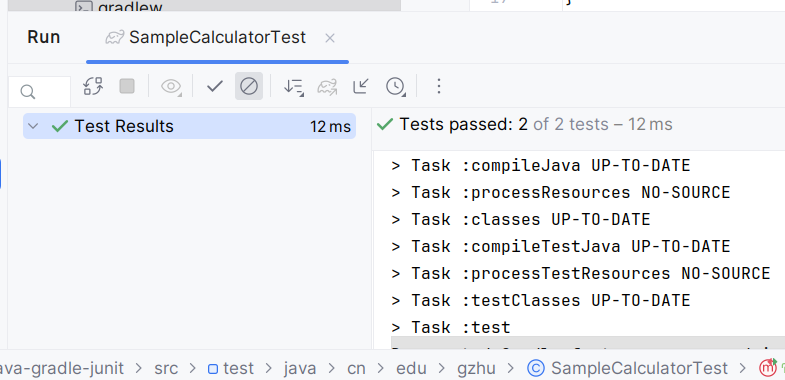
添加业务逻辑。



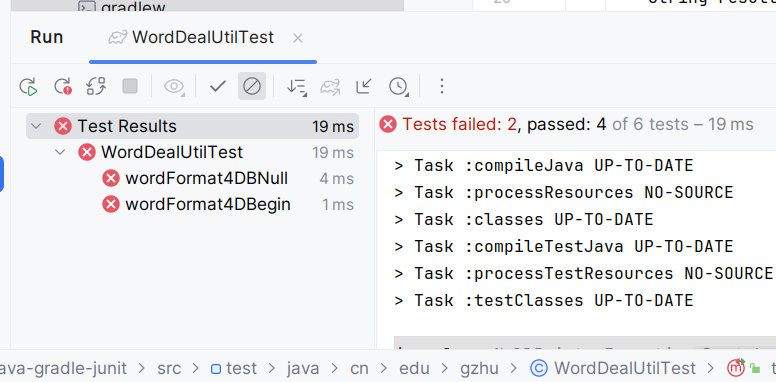
针对其添加测试（到test源文件根）。



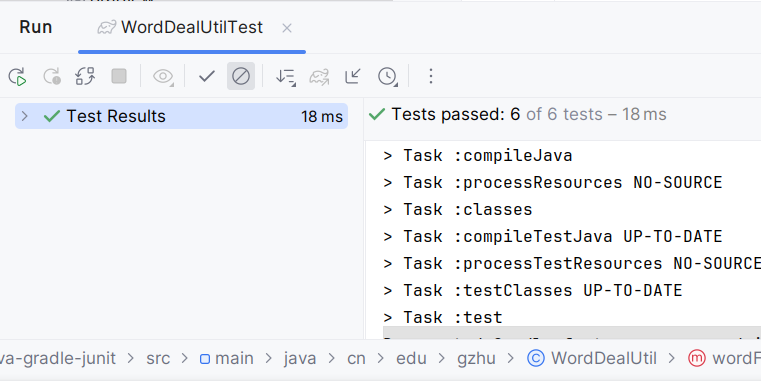
运行测试，通过。



按照要求添加WordDealUtil类，添加单元测试，运行通过。按照要求更改完善测试之后，运行出现错误。

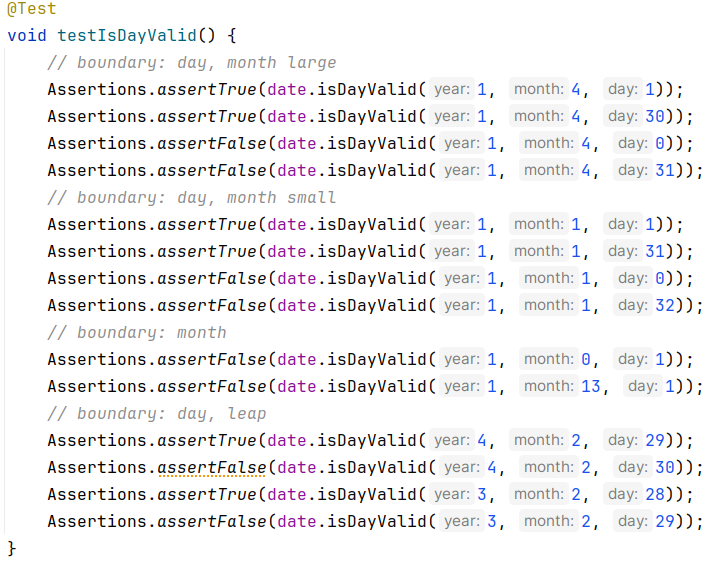


修复漏洞之后，测试通过。



2、使用Junit框架对类Date和类DateUtil进行单元测试。

对闰年、平年分别测试大月小月的边界值，还有2月的边界值，发现倒数第三个assertion出现问题。



源代码中发现逻辑错误，予以修改，通过测试。



测试1到12月的有效性，和两个无效边界值。年份测试有效年边界值和无效年边界值。

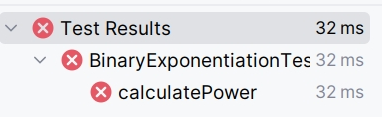


3、使用Junit框架对类Heap\_Item和类BinaryExponentiation进行单元测试。

对BinaryExponentiation进行测试，编写以下测试用例（两个函数测试方法同理，故不赘述，仅展示其一）：

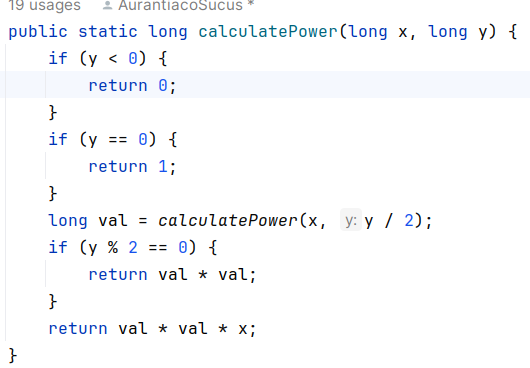


经过测试发现负指数的情况出现了结果不正确的情况。

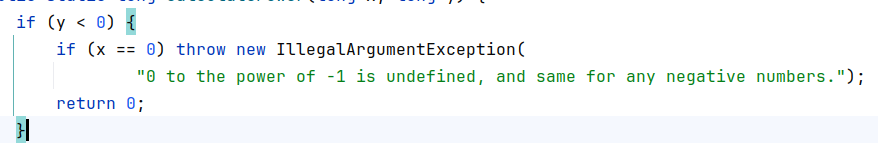


修改代码，添加负指数逻辑。可以推测负指数情况下这两个返回整型值的函数只可能返回0（小数被舍弃）。

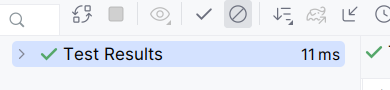
另一个函数的修改同理。修改完成再次运行测试，发现仍然不通过。在输入不合法的情况下没有抛出错误。（无关部分已略过）



再次修改代码，添加相关判断。



修改后再次测试，所有测试用例通过。



4、完成以下用例设计

A、采用决策表法完成“隔一天日期问题”的测试用例设计

给出输入变量month、day、year划分的有效等价类；画出决策表；设计测试用例

其它实验：此处给出实验过程中遇到的问题（如果有的话），或给出测试运行结果

有效等价类分析如下：

month

* 一年有12个月：0<month<13

day

* 对于大月，一个月有31天：0<day<32
* 对于小月，一个月有30天：0<day<31
* 对于闰年的2月，一个月有29天：0<day<30
* 对于平年的2月，一个月有28天：0<day<29

year

* 正四位数：999<year<10000

B、采用成对测试方法进行配置测试

1）、给定基本配置情况，采用工具设计测试用例

此处给出主要的测试用例

2）、变换或增加配置约束，观察测试用例的变化

此处分析测试用例的变化情况

C、基于独立路径的测试用例设计

画出这段代码的控制流图；计算环形复杂度；列出独立路径；设计测试用例。

D、基于逻辑覆盖的测试用例设计

将源代码转化为程序流程图；按照不同的覆盖准则（包括语句覆盖、判定覆盖、条件覆盖、判定-条件覆盖、条件组合覆盖）设计测试用例。

五、实验结果（或实验代码）

对于实验2、实验3编写了测试代码或使用了测试工具进行测试的，给出相应的测试代码和测试运行结果

六、问题与体会

此处给出实验中遇到的问题和自己的体会