**面向对象程序设计基础 第四次作业**

徐浩博 2020010108

**模型部分**

* 公历历法

目前我们沿用的公历是1582年教皇格里高利十三世设置的格里高利历(Gregorian Calendar)，该历法结合当时观测的平均回归年365.2425天的估值进行设置，规定每四年一闰，但逢百年不闰，而逢四百年又再闰. 利用现代科学技术手段，科学家观测出更精确的平均回归年长度365.242199074天，发现运行千年后，现行格里高利历的误差将相当可观，于是提出了许多修正方案，这里我们采用其中一种，将格里高利历的修正历作为此日期计算器的历法. 此历法规定：

每4年1闰，100年不闰，400年再闰，3200年不闰，172800年再闰.

虽然经过十万年数量级的时间后，此历法又将出现较大误差，但根据已有的观察，地球公转速度在逐渐改变，因此待到那时，也许将会有新的历法代替现有历法，因此我最终还是采用了这种格里高利历的修正历.

* 类的说明

实现本任务，我共设计了三个类，分别为integer, calendar, calendarmain，下面我将逐一分别说明：

* integer类：设置变量num，设置read()函数，读入一个整数（可以识别负整数）并将此整数赋值给num.
* calendar类：设置yy, mm, dd，分别对应储存的年月日的值. data数组储存平年每月的天数，data\_数组储存闰年每月的天数. 函数init()将系统的日期数据赋值给年月日变量. set(int,int,int)函数，调用时可以传递三个参量，分别对应年月日，可以将需要设置的年月日赋值给类体的年月日变量. cal(int)函数，参量表示要计算n天后的日期，0表示当天，负整数表示计算n天前，并将计算得结果直接输出.
* calendarmain类：仅设置calendar类储存日期并通过函数调用完成相关功能. 函数run()相当于一个总控函数，可以对输入的命令进行处理并完成相关操作.

**验证部分**

**等价类划分**

* 首先将integer的read函数
* 其次可以测试calendar类的init函数
* 再次可以测试calendar类的set函数
* 再次测试calendar类的cal函数

**测试方法与结果**

* 测试integer的read函数

另写main文件，定义integer类并调用read函数，再将读入的值输出比照二者是否一致.

|  |  |
| --- | --- |
| 输入 | 输出 |
| 1 | 1 |
| 0 | 0 |
| -1 | -1 |
| 7346468 | 7346468 |
| -7346468 | -7346468 |

* 测试calendar类的init函数

因为在calendar的构造函数中已调用init函数，因此可以对现在日期进行直接验证，从而达到验证init函数正确性的目的.

输入指令时，输入计算0天后的指令，从而可以对init函数进行检验.

|  |  |
| --- | --- |
| 输入 | 输出 |
| N/A | 公元2021年3月21日 |

* 测试calendar类的set函数

输入指令时，可以先设置一个时间，再在输入指令时，输入计算0天后的指令，从而可以对set函数的正确性进行检验.

|  |  |
| --- | --- |
| 输入 | 输出 |
| 2020/3/1 | 公元2020年3月1日 |
| 100/10/25 | 公元100年10月25日 |
| -1/1/1 | 公元前1年1月1日 |
| 1949/10/32 | 输入有误 |
| 2021/13/1 | 输入有误 |

* 测试calendar类的set函数

set函数是本任务中最容易出错的部分，因此我们进行如下等价类进行测试，特别地，我们对每个等价类中是否有闰二月进行考虑.

下面的测试样例均先复制再进行日期的计算.

1.不跨月的计算

|  |  |
| --- | --- |
| 输入 | 输出 |
| 2020/2/28 1 | 公元2020年2月29日 |
| 2020/2/15 -14 | 公元2020年2月1日 |
| -1/1/1 1 | 公元前1年1月2日 |
| -1/12/15 -14 | 公元前1年12月1日 |
| 2021/3/1 0 | 公元2021年3月1日 |

2.跨月但不跨年的计算

|  |  |
| --- | --- |
| 输入 | 输出 |
| 2020/2/29 1 | 公元2020年3月1日 |
| 2020/2/15 -16 | 公元2020年1月31日 |
| -1/1/1 100 | 公元前1年4月11日 |
| 2004/3/21 -29 | 公元2004年2月21日 |

3.跨年的计算

|  |  |
| --- | --- |
| 输入 | 输出 |
| 2000/3/21 -365 | 公元1999年3月21日 |
| 2020/1/1 366 | 公元2021年1月1日 |
| 500/8/5 1000 | 公元503年5月2日 |
| 500/8/5 -1000 | 公元497年11月8日 |

4.跨BC/AD的计算

|  |  |
| --- | --- |
| 输入 | 输出 |
| 1/1/1 -3 | 公元前1年12月29日 |
| -1/12/1 32 | 公元1年1月2日 |
| -2/5/1 1000 | 公元前2年1月26日 |

* 测试calendarmain类的run函数

此测试主要针对于不同命令一次执行是否正确，因此，可将以上不同命令任意选取几组进行测试.

测试样例同上，不再赘述.

参考文献

[1]W. Bulmer & W .Nicol. An Illustraion of the Liturgy[M]. London: Library of the University of Virginia. P.110-112.

[2]C.F. Marvin. Leap years rules and calendar accuracy[J]. Popular Astronomy, Vol.31. P.298-308.