### OOP第四周作业文档

2019010175 孔瑞阳 土木92

### 项目信息

**1、功能说明**

先询问是否使用系统当前时间，如果不使用输入一个合法的年月日（公历）。

再输入一个整数n。

输出输入的日期经过n天后的日期。

### **软件构件介绍**

|  |  |
| --- | --- |
| **文件** | **功能介绍** |
| CP\_Integer.h/cpp | 实现整数的输入和操作 |
| CP\_Calendar.h/cpp | 日历类，实现日历内部的操作 |
| CP\_CalendarApplication.h/cpp | 日历的交互操作 |
| CP\_CalendarTest.h/cpp | 日历类的测试程序 |
| calendarMain.cpp | 主程序 |

### **3、测试环境**

|  |  |
| --- | --- |
| CPU | Intel(R) Core(TM)i7-9750H CPU @ 2.6Ghz 6核12线程 |
| GPU | NVIDIA GeForce RTX2070 |
| RAM | DDR4 16G+16G |
| Operating System | Microsoft Windows 版本1809 (OS 内部版本17763.1098) |
| Compiler | MSVC++ 14.24 |

### 模型

##### 输入与输出

输入和输出采用C++标准库中的输入输出流cin和cout进行。

在输入过程中，

1. 没有公历0年，若年份输入0则重新输入。
2. 月份的范围是1-12，如果输入的月份不在这个范围内重新输入。
3. 根据之前输入的年份和月份，可以推出这个月的天数，得出日期的范围。如果输入的日期不在计算的范围内则重新输入。

对于输出，

1、如果年份x是正数，则输出公元x年

1、如果年份x是负数，则输出公元前|x|年

##### 公元历法

参考链接：<https://en.wikipedia.org/wiki/Gregorian_calendar>

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **月份** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **天数** | 31 | 28(闰年29） | 31 | 30 | 31 | 30 | 31 | 31 | 30 | 31 | 30 | 31 |

闰年规则：如果年份是4的倍数是闰年，但是如果是100的倍数就不是闰年，其中又如果是400的倍数是闰年。

##### 日期计算

首先，将每一个日期和一个整数一一对应，这个整数代表公元前1年12月31日到这个日期的天数。比如，公元1年1月1日对应1，公元前1年12月31日对应0，公元前1年12月30日对应-1。

那么我们只要实现两个操作：

1. 把一个日期转换为一个整数。
2. 把一个整数转换成一个日期。

把输入的日期转换为整数之后，进行“n天后”的这个操作只要直接做加法就可以了，再把+n之后的整数转换为日期。

具体来说：

**(1)判断闰年**

对于公元后x年，则根据闰年规则计算。

对于公元前x年，将x-1进行上述操作即可。（如B.C.1是闰年，B.C.101不是，B.C.401是闰年）

1. **日期转化为整数**

**第一步：计算本年份之前的年份有多少天**

由于公历严格按照400年为一个周期，每400年根据容斥原理有100-4+1=97个闰年，所以400年的总天数永远是365\*400 + 97 = 146097天。

那么我们可以先计算年的天数（公元前直接转化为绝对值即可），总天数就是，再从第年循环到第y-1年，每遇到一年通过判断这一年是否是闰年来计算当年的天数。可以看出这个循环不超过400次。

**第二步：计算本月份之前的月份有多少天**

如果是公元后，就从1月循环到m-1月，否则从m+1月循环到12月。如果遇到2月，通过判断当年是不是闰年来决定天数。

**第三步：计算本月份有多少天**

如果是公元后，那么日期d直接是天数。

如果是公元前，根据年份和月份计算出当月的天数，那么结果就是这个月剩下的天数。

1. **整数转化为日期**

**第一步：计算年份**

首先，还是按照400年的周期先计算自己应该处于哪一个400年，初始化年份为（如果是公元前则取负）,并减去相应的天数。其次进入一个循环，如果当前剩余的天数比现在的年份还要大的话，就减去当前年份的天数，并把当前年份+1（如果是公元前则-1），直到当前剩余的天数小于本年份的天数。可以看出这个循环不会超过400次。

**第二步：计算月份**

如果是公元后则从1月开始向后循环，如果是公元前则从12月开始向前循环。

每循环到一个月份，计算这个月份有多少天，如果当前剩余的天数大于这个月的天数，则减去这个月的天数继续循环，否则当前循环到的月份就是最终的月份。

**第三步：计算日期**

如果是公元前，则当前剩余的天数D就是日期。

如果是公元前，那么从这个月的最后一天往前推D-1天即可。

### 单元测试

##### 设置系统时间的测试

测试时间：2020年3月15日

是否选择系统时间选择yes，输出为 日期是：公元2020年3月15日

##### 输入单元的测试

**对于输入年份**

1. 等价类划分
2. 输入的不是整数
3. 输入为0
4. 输入一个不为0的整数
5. 测试：

|  |  |  |
| --- | --- | --- |
| **等价类** | **选取案例** | **输出结果（下一步操作）** |
| 等价类① | HelloWorld! | 您输入的不是整数，请输入一个整数： |
| 1919.810 | 您输入的不是整数，请输入一个整数： |
| 等价类② | 0 | 年份不能为0，请重新输入 |
| 等价类③ | 1919810 | 请输入月份 |
| -114514 | 请输入月份 |
| 1 | 请输入月份 |
| -1 | 请输入月份 |

**对于输入月份**

1. 等价类划分

① 输入的不是1-12的正整数。

② 输入的是1-12的正整数。

1. 测试：

|  |  |  |
| --- | --- | --- |
| **等价类** | **选取案例** | **输出结果（下一步操作）** |
| 等价类① | HappyWorld! | 您输入的不是整数，请输入一个整数： |
| 114.514 | 您输入的不是整数，请输入一个整数： |
| -10 | 您输入的月份不在1-12的范围内，请重新输入 |
| 100 | 您输入的月份不在1-12的范围内，请重新输入 |
| 等价类② | 1 | 请输入日期（用1-31的整数表示） |
| 4 | 请输入日期（用1-30的整数表示） |
| 9 | 请输入日期（用1-30的整数表示） |
| 10 | 请输入日期（用1-31的整数表示） |
| 2 (2020年) | 请输入日期（用1-29的整数表示） |
| 2 (-2021年) | 请输入日期（用1-29的整数表示） |
| 2 (1年) | 请输入日期（用1-28的整数表示） |

**对于输入日期**

（1）等价类划分

① 输入的不是整数

② 输入的日期不在指定范围内

③ 输入的日期合法

（2）测试：

|  |  |  |
| --- | --- | --- |
| **等价类** | **选取案例** | **输出结果（下一步操作）** |
| 等价类① | HappyWorld! | 您输入的不是整数，请输入一个整数： |
| 114.514 | 您输入的不是整数，请输入一个整数： |
| 等价类② | 31 （4月） | 您输入的日期不在1-30的范围内，请重新输入 |
| -1 （4月） | 您输入的日期不在1-30的范围内，请重新输入 |
| 29 （1年2月） | 您输入的日期不在1-28的范围内，请重新输入 |
| 0 （7月） | 您输入的日期不在1-31的范围内，请重新输入 |
| 等价类③ | 29 （-1年2月） | 请输入一个整数表示n天后 |
| 31 （5月） | 请输入一个整数表示n天后 |
| 1 （3月） | 请输入一个整数表示n天后 |
| 31 （8月） | 请输入一个整数表示n天后 |

**对于输入n天后**

由于只涉及到整数的输入，在上述测试中已经得到验证。

##### 设置、输出时间；获取年月日单元的测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入时间 | showYear | showMonth | showDay | outputCalendar |
| 2020/3/15 | 2020 | 3 | 15 | 日期是：公元2020年3月15日 |
| -1/6/1 | -1 | 6 | 1 | 日期是：公元前1年6月1日 |
| 2017/3/14 | 2017 | 3 | 14 | 日期是：公元2017年3月14日 |
| -124353/2/29 | -124353 | 2 | 29 | 日期是：公元前124353年2月29日 |

##### 4、日期与天数转换单元的测试

测试单元：**CP\_CalendarTest.h/cpp**

**第一步：利用在线日历网站进行日期转换为天数的测试（公元后）。**

日历网站：<http://tools.jb51.net/static/api/datecalc/index.html>

将起始日期调成0001-01-01，将结果+1即可。

测试结果：

|  |  |  |
| --- | --- | --- |
| **日期** | **日历网站** | **程序运行结果** |
| 9999-12-31 | 3652059 | 3652059 |
| 0001-01-01 | 1 | 1 |
| 0114-05-14 | 41406 | 41406 |
| 1919-08-10 | 700756 | 700756 |
| 2020-03-14 | 734798 | 734798 |

两者相同，初步测试成功。

**第二步：利用相互转换自动测试。**

1. 不断地随机一个天数，将其转换为日期，再转换为天数，判断是否相同。
2. 不断地随机一个日期，将其转换为天数，再转换为日期，判断是否相同。

每分钟可以进行上千万次运算，在经过10分钟上亿次运算之后依然没有出现错误，可以判断基本正确。

##### 5、n天后日期的测试

具体的实现正确性已经在 4、日期与天数转换单元测试 中得到验证。

所以仅验证特殊/边界情况。

|  |  |  |
| --- | --- | --- |
| **输入日期** | **n** | **程序运行结果** |
| 1/1/1 | -2 | n天后的日期是：公元前1年12月30日 |
| -1/12/31 | 10 | n天后的日期是：公元1年1月10日 |
| -1/12/31 | 2147483647 | n天后的日期是：公元5879611年7月11日 |
| -1/12/31 | -2147483647 | n天后的日期是：公元前5879611年6月23日 |
| -1/12/31 | -2147483648 | n天后的日期是：公元5879599年12月3876日 |
| 1/1/1 | 2147483647 | n天后的日期是：公元5879599年12月3876日 |
| 2/1/1 | 2147483647 | n天后的日期是：公元前5879610年6月22日 |

通过测试，特殊情况（公元前后转换）无误，但是由于int totalDays范围的限制，

能表示的日期为：公元前5879611年6月23日 - 公元5879611年7月11日。