实验：C++综合实验

1. 实验目的

a) 进一步培养学生面向对象程序设计的思想，加深对高级语言基本语言要素和控制结构的理解。

b) 针对 C++语言中的重点和难点内容进行训练，要求独立完成有一定工作量的程序设计任务。

2. 实验环境

1) Ubuntu 18.04 LTS

2) G++

3. 实验内容

设计一个日期类 MyDate

4. 实验步骤

1、设计一个日期类 MyDate (1)

(1) 提供转换构造函数，把整数转成日期，规则为整数代表自 1970-01-01 00:00:00 UTC 起经过的秒数。

(2) 提供构造函数,接受三个整数分别代表年、月、日三个字段。

(3) 提供成员函数 bool equals(const MyDate&) const 当二个对象年月日均相等时返回 true，否则返回 false 。

提供成员函数 bool lessThan(const MyDate&) const,当此对象的日期早于形参日期时返回 true,否则返回 false。

类似的,提供以下的成员函数:

bool lessThanOrEquals(const MyDate&) const

bool greaterThan(const MyDate&) const

bool greaterThanOrEquals(const MyDate&) const

(4) 提供成员函数 MyDate add(int) const,意味着返回此对象增加相应天数的日期。类似的,提供以下成员函数 MyDate subtract(int) const。

(5) 提供成员函数 int subtract(MyDate) const,返回值为此对象与形参之间的距离,当此对象比形参晚时返回正整数。

(6) 提供成员函数 string toString() const,输出符合 ISO8601 标准。

2、设计一个日期类 MyDate (2)

(1) 检查第 1 版中是否能支持 UNIX epoch 之前的日期。

(2) 检查第 1 版中是否支持 epoch+2147483647 秒(代表格林尼治时间 2038 年 1 月 19 日凌晨 03:14:07)之后的日期,以及是否支持 epoch-2147483648 秒(代表格林尼治时间 1901 年 12 月 13 日 20:45:52)之前的日期

(3) 检查第 1 版中比较日期时采用的是 UTC 还是本地时间。

(4) 第 2 版要求支持公元前 9999 年 1 月 1 日至公元 9999 年 12 月 31 日,公元前年份的表达方式遵守 ISO8601 标准:公元 1 年为 1,公元前 1 年为 0,公元前 2 年为-1,以此类推。

(5) 第 2 版要求比较日期和显示日期时采用本地时间(北京时间)

(6) 在公元 1752 年 9 月 2 日(周三)之前使用儒略历,其后一天起使用格里历,调整为 1752 年 9 月 14 日。星期则连续,2 日为周三,14 日为周四。

3、设计一个日期类 MyDate (3)

(1) 提供异常处理。对非法数据和操作,需要抛出异常。

例如,对构造函数 MyDate(int,int,int)的情况,要进行输入合法性判断,如果输入不合法,则需要抛出异常:throw invalid_argument("Invalid Date!");

(2) 以下运算符重载为成员函数:<, <=, >, >=, ==。

功 能 与 lessThan, lessThanOrEquals, greaterThan, greaterThanOrEquals,

equals 相同。可以直接调用上述函数。

(3) 以下运算符重载为成员函数:重载二元+,-。

功能与 add(int days), subtract(int days)相同。可以直接调用上述函数。只需要考虑形如 myDate+5 这种情况,不考虑 5+myDate。

功能:可以对一个日期加/减一个整数,整数的含义为天数。

(4) 以下运算符重载为成员函数:重载二元-。

功能与 subtract(const MyDate&)相同。

功能:二个日期类之间可以做减法,结果为相差的天数,

(5) 重载<<以便使用 cout 输出。

功能是将 toString()的返回值直接输出即可。

输出格式为 ISO8601 标准,即形式为:2017-01-01;公元前 1 年记为 0,形式为:0000-01-01;公元前 2 年记为-1,形式为:-0001-01-01。

4、设计一个日期类 MyDate (4)

(1) 将儒略历和格里历的切换时间改成可修改的。增加一个设置国家/地区的函数 static void setLocale(string locale)。这个参数可以影响儒略历/格里历的切换时间。为简化起见,暂时可以设置的参数 country 值有:

US:美国,历法切换时扣除 1752-09-03~1752-09-13(11 天)

IT:意大利,历法切换时扣除 1582-10-05~1582-10-14(10 天)

例如在 main()中:setLocale(“IT”);

(2) 考虑时区问题。修改类定义,取消 public 常量 TIMEZONE_OFFSET,并增加函数 static void setTimeZoneOffset(long long offset)。

例如在 main()中:setTimeZoneOffset(3600LL*8); //北京时间东 8 区。

(3) 增加函数 int getWeekday(),获得当前对象的星期数。返回值的含义:0 代表周日,1 代表周一.....6 代表周六。已知 1970 年 1 月 1 日是星期四。

5. 实验方式

分组开展课程设计；每组同学合作上机编程实验，实验指导教师现场指导。

a) 团队合作设计，撰写软件工程文档，完成设计、开发和测试，分组汇报及答辩。

b） 能够使用 Git 等代码管理工具，进行协同开发；

6. 参考内容

1）《C++程序设计（原书第 3 版）》，[美] Y.Daniel Liang 著；刘晓光，李忠伟，任明明 等 译，机械工业出版社，2015.01

2）《C++ Primer Plus（第 6 版 中文版）》，[美] Stephen Prata 著；张海龙，袁国忠 译，人民邮电出版社，2012.06

7. 相关软件下载

见课程网盘

8. 实验报告要求

1) 项目代码

2) 项目组分工说明书