Work6 继承和派生（1）

2024.4.3

作业上交时请按照这种方式命名文件：学号+姓名+work6.cpp

命名时，请一定把学号写在姓名前面。这样收到的作业容易排序。

如果需要上交多个文件： 学号+姓名+work6.rar （其它压缩格式也可以）

最晚上交时间： 2024.4.10

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 夏威夷 | 阿拉斯加 | 洛杉矶 | 芝加哥 | 纽约 | | | 大西洋时间 | 巴西 | 大西洋-中部 | | |
| Hawaii | Alaska | LosAngeles | Chicago | NewYork | | |  | Brazil |  | | |
| ha | ak | la | ch | ny | | | dxy | bz | dxyz | | |
| GMT-10 | GMT-9 | GMT-8 | GMT-6 | GMT-5 | | | GMT-4 | GMT-3 | GMT-2 | | |
|  |  |  |  | |  |  | |  | |  |  | |
| 伦敦 | 柏林 | 开罗 | 莫斯科 | | 新德里 | 曼谷 | | 北京 | | 东京 | 悉尼 | |
| London | Berlin | Cario | Moscow | | NewDelhi | Bangkok | | Beijing | | Tokyo | Sydney | |
| ld | bl | ca | mo | | nd | bk | | bj | | tk | sy | |
| GMT | GMT +1 | GMT +2 | GMT +3 | | GMT +5 | GMT +7 | | GMT +8 | | GMT +9 | GMT +10 | |

1. 时间-世界时间类
   1. 基类 Clock

成员数据（私有）：int \_h, \_m, \_s 表示小时、分钟、秒

成员函数（公有）：

构造函数：

时分秒的默认值都是0

标准化函数：void Standard();

小时的取值范围0~23 ，分钟和秒的取值范围0~59

Set函数

Get函数（常函数）

显示函数（常函数）

求2个时间值的差函数：Clock Sub(const Clock& c) const;

其他函数自行设计

* 1. 派生类 WorldClock （世界时间） 公有继承Clock类

增加成员数据（私有）：

string \_city

增加成员函数（公有）：

构造函数

默认值北京时间Beijing 0:0:0

Set函数

Get函数（常函数）

显示函数（常函数）

求2个时间值的差函数：WorldClock Sub(const World Clock& c) const;

（派生类的成员函数要尽量地调用基类的成员函数）

其他函数自行设计

* 1. 主函数

实现2个功能：时间转换 和 时间差

参看1.4程序的一次输入输出

选项： C(c) Conversion 表示时间转换

S(s) Subtraction 表示求时间差

E(e) End 表示程序结束

1.4 程序的一次输入输出（供参考）

Conversion / Subtraction / End : c

Input City: bj

Input hour minute second: 20 30 45

Input Conversion City: sy

Time is Bejing 20:30:45

Sydney 22:30:45

Conversion / Subtraction / End : s

Input City: bj

Input hour minute second: 20 30 45

Input City: ld

Input hour minute second: 10 25 5

Beijing 20:30:45 – London 10:25:05 = Beijing 02:05:40

Conversion / Subtraction / End : c

Input City: ca

Input hour minute second: 8 12 50

Input Conversion City: ha

Time is Cario 08:12:50

Hawaii 20:12:50

Conversion / Subtraction / End : s

Input City: bk

Input hour minute second: 15 26 30

Input City: ny

Input hour minute second: 18 55 5

Bangkok 15:26:30 – NewYork 18:55:05 = Bangkok 08:31:25

Conversion / Subtraction / End : e

Byebye~

1.5提示和要求：

要求：

WorldClock 继承 Clock

WorldClock的成员函数要尽量地调用Clock相应的成员函数

使用多文件组织

提示：

使用枚举表示城市

使用全局数组表示城市与时区的关系

1. 选做题

时钟-闹钟类

在时钟类Clock的基础上**派生**闹钟类AlarmClock

（1）时钟类Clock

Clock类设计如下，请实现成员函数。

class Clock

{ private:

int \_h,\_m,\_s; //24小时制的时间数据

public:

Clock(int hour, int minute, int second);

void Show() const; //显示时间功能

void Tick(); //计时功能，每一次Tick，时间向前走1秒

void Set(int h, int m, int s); //重新设定时间值

};

（2）闹钟类AlarmClock

功能基本要求：

**闹钟类继承时钟类**，闹钟的功能包括**计时**和**闹铃**。

* 设计AlarmClock(闹钟)类：

a) 增加表示闹铃时间的数据成员，闹铃时间精确到小时 : 分钟；

b) 构造函数构造新闹钟对象，初始化时钟时间和闹铃时间；

c) 成员函数Alarm()的功能是：当闹铃时间到时，增加实际的声音效果“嘀嘀嘀……”；

d) 成员函数SetAlarmTime(…)用来重新设置闹铃的时间值；

e) 重写Tick()函数，保持计时功能，增加判定是否到达闹铃时间的功能，如果到达闹铃时间则响铃（Alarm()）；

（3）主程序

使用AlarmClock类

a)创建AlarmClock类的对象，给它初始时间值（比如：23：59：30），闹铃时间值（比如： 00：01 表示0点1分）和它要走的时长（比如：2分15秒），发出启动命令后，让它一秒一秒地走起来，并且显示它的每个时间值（包含时：分：秒）

b)当到达闹铃时间时，增加实际的声音效果“嘀嘀嘀……”。

c)等到预订的时长完成时，输出计时结束信息

d)以上的计时过程可以重复多次

（4）输入输出样例：

请输入初始时间（时 分 秒）：23 59 45

请输入闹铃时间（时 分） ：0 1

请输入计时时长（分 秒） ：2 30

23:59:45 //（这里在相同的位置显示从23时59分45秒到0时2分15秒）

//(其中，在00:01:00到00:01:59期间，增加声音“嘀嘀嘀......” )

计时结束，还要继续吗（Y/N）？Y

请输入初始时间（时 分 秒）：2 5 3

请输入闹铃时间（时 分） ：2 6

请输入计时时长（分 秒） ：2 15

02:05:03 //（这里在相同的位置显示从2时5分3秒到2时7分18秒）

//(其中，在02:06:00到02:06:59期间，增加声音“嘀嘀嘀......” )

计时结束，还要继续吗（Y/N）？Y

…….. //（以上过程重复）

计时结束，还要继续吗（Y/N）？N

Press any key to continue

(5) 本程序请使用多文件组织

clock.h clock.cpp

alarm.h alarm.cpp

main.cpp

(6)编程提示：

（a）字符'\7' ('\a'响铃)可以让计算机发出“嘀”声，例如：

cout << '\7' << '\7' << '\7';

（b）字符’\r’ (回车)可以控制输出的内容在同样的位置

函数“Sleep()”可以让程序暂停，参见以下代码

#include <iostream>

#include<windows.h>

using namespace std;

int main()

{ for(int i=0;i<10;i++)

{ cout<<i<<" "<<'\r';

Sleep(1000);

}

return 0;

}

(c)设置输出格式的相关操作:

setw(int)可以设置输出内容的宽度;

setfill(char)可以设置前导符号;

这些格式设置操作的使用要求包含头文件<iomanip>

例：

char s[5]=”abc”;

cout<<setw(10)<<setfill('%')<<s<<endl;

将输出：

%%%%%%%abc