

Homework 8 Ch7 继承与派生

2018年11月06日

1. 时钟类-闹钟类设计（必做）

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 Hour, Minute, Second;           //24小时制的时间数据
public:
    Clock(int hour, int minute, int second);
    void ShowTime();                   //显示时间功能
    void Tick();                       //计时功能，每一次Tick，时间向前走1秒
    void SetTime(int NewH, int NewM, int NewS); //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：

- 增加表示闹铃时间的数据成员，闹铃时间精确到小时：分钟；
- 构造函数构造新闹钟对象，初始化时钟时间和闹铃时间；
- 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“滴滴滴.....”；
- 成员函数SetAlarmTime(...)用来重新设置闹铃的时间值；
- 重载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。

- 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）。
- 当到达闹铃时间时，增加实际的声音效果“滴滴滴.....”。

- 等到预订的时长完成时，输出计时结束信息。
- 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（回车）
请输入闹铃时间（时 分）：0 1（回车）
请输入计时时长（分 秒）：2 30（回车）
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“滴滴滴.....”)
计时结束，还要继续吗（Y/N）？Y（回车）
请输入初始时间（时 分 秒）：2 5 3（回车）
请输入闹铃时间（时 分）：2 6（回车）
请输入计时时长（时 分）：2 15（回车）
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“滴滴滴.....”)
计时结束，还要继续吗（Y/N）？Y（回车）
.....      //(以上过程重复)
计时结束，还要继续吗（Y/N）？N（回车）
Press any key to continue
```

(5) 本程序请使用多文件组织

各个文件如下：

- clock.h clock.cpp
- alarm.h alarm.cpp
- main.cpp

(6) 编程提示：

- 字符'\7' (响铃)可以让计算机发出“嘀”声，例如：

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

- 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;
}
```

- 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`。

2. 校园人员类设计（必做）

一个学校的人员管理系统中，有学生、员工、助工三种类型的人,助工由勤工俭学的学生来兼任，兼有学生和员工两类人的特征，设计合适的类实现此人员管理系统。

（1）设计Person类

包括成员：

- 成员变量：name
- 成员变量：age
- 成员函数：SetName()
- 成员函数：GetName()
- 成员函数：SetAge()
- 成员函数：GetAge()

（2）设计Student类

继承于Person类，增加了新成员：

- 成员变量：Major
- 成员变量：StudentID
- 成员函数：SetMajor()
- 成员函数：GetMajor()

- 成员函数：SetStudentID()
- 成员函数：GetStudentID()

(3) 设计Employee类

继承于Person类，增加了新成员：

- 成员变量：Department
- 成员变量：EmployeeID
- 成员函数：SetDepartment()
- 成员函数：GetDepartment()
- 成员函数：SetEmployeeID()
- 成员函数：GetEmployeeID()

(4) 设计StuEmployee类

继承于Student和Employee类，无新增成员。

(5) 编写main函数测试

完成以下样例的输入与显示：

```
学生兼职人员输入系统：
请输入学生姓名：张三
请输入学生年龄：20
请输入学生专业：计算机
请输入学生学号：201511210111
请输入学生工号：20181106001
请输入工作院系：计算机系
您输入的学生兼职人员信息如下：
姓名：张三
年龄：20
专业：计算机
学号：201511210111
工号：20181106001
院系：计算机系
```

(6) 使用多文件组织。

- student.h student.cpp
- employee.h employee.cpp
- stuemployee.h stuemployee.cpp
- main.cpp

3. 日期时间类设计（选做）

(1) 设计日期类CMyDate。描述年、月、日，设计构造函数、Set函数、Next函数（表示明天）、Display函数等。

(2) 设计**时间类CMyTime** . 描述小时、分钟、秒，设计构造函数、Set函数、Next函数（表示下1秒）、Display函数等。

(3) 设计**日期时间类CMyDateTime**, **日期时间类继承日期类和时间类**。日期时间类的数据有：年、月、日、小时、分钟、秒，设计构造函数、Set函数、Next函数（表示下1秒）、Display函数等。

(4) 输入输出样例：

开始日期时间类的测试：

```
请输入日期时间值（年月日时分秒）：2015 5 6 10 29 30
现在的日期时间值是：2015-05-06 10:29:30
下一个日期时间值是：2015-05-06 10:29:31
继续吗？（Y/N）：Y
请输入日期时间值（年月日时分秒）：2015 5 6 10 29 59
现在的日期时间值是：2015-05-06 10:29:59
下一个日期时间值是：2015-05-06 10:30:00
继续吗？（Y/N）：Y
请输入日期时间值（年月日时分秒）：2015 5 6 23 59 59
现在的日期时间值是：2015-05-06 23:59:59
下一个日期时间值是：2015-05-07 00:00:00
继续吗？（Y/N）：Y
请输入日期时间值（年月日时分秒）：2015 5 31 23 59 59
现在的日期时间值是：2015-05-31 23:59:59
下一个日期时间值是：2015-06-01 00:00:00
继续吗？（Y/N）：Y
请输入日期时间值（年月日时分秒）：2015 12 31 23 59 59
现在的日期时间值是：2015-12-31 23:59:59
下一个日期时间值是：2016-01-01 00:00:00
继续吗？（Y/N）：N
Press any key to continue.....
```

(5) 本程序使用多文件组织
如：

- mydate.h mydate.cpp
- mytime.h mytime.cpp
- mydatetime.h mydatetime.cpp
- testdatetime.cpp