类与对象（一）——类与对象的定义

> 张林鹏_2021032449

一、实验目的

1. 熟悉类的构成，掌握类的定义方法；
2. 掌握对象的定义及对象成员的访问方法；
3. 初步熟悉类与对象简单应用及编程。

实验内容

程序1: exp_101.cpp

1. 程序的输出结果为:

```
real of complex A=3
imag of complex A=4
abs of complex A=5
```

2. 成员函数`set_complex(double r,double i)`的作用是设置实部和虚部的值

   `get_real( )`的作用是计算实部的值.

   `get_abs()`的作用是计算该复数的模.

3. 将main()函数中的语句行`cout <<A.get_real()<<endl`改为`cout <<A. real<<endl`,重新编译程序,将出现 编译错误 , 其原因是 real 是私有成员变量, 不能直接在类外部访问.

程序2: exp_102.cpp

1. 你分析的输出结果是:

```
x=20  y=30
x=20  y=30
x=25  y=35
```

程序的实际结果是:

```
x=20  y=30
x=20  y=30
x=25  y=35
```

程序3

1. 完善`hdata.h`中类的定义:

```
class Date
{
private:
    int year,month,day;
public:
    void set_date(int y=2000,int m=1,int d=1)  //对数据成员赋值
    {
        year = y;
        month = m;
        day = d;
    }
    int get_year()                          //返回year
    {
        return year;
    }
    int get_month()                        //返回month
    {
        return month;
    }
    int get_day()                          //返回day
    {
        return day;
    }
    int isleapyear(void);                   //是闰年返回1,不是闰年返回0
    void print_date(void)
    {
        cout<<year<<' -' <<month<<' -' <<day<<endl;
    }
};
int Date::isleapyear(void)                  //是闰年返回1,不是闰年返回0
{
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
        return 1;
    else
        return 0;
}
```

2. 按注释要求完善下列程序(exp_103.cpp)

```
#include <iostream.h>
#include"hdate.h"
void main(void)
{
    Date da1, da2;
    int y, m, d;
    da1.set_date(2004, 5, 1);
    da1.print_date();
    cout <<"year =";
    cin >> y;
    cout <<"month =";
    cin >> m;
    cout <<"day =";
    cin >> d;
    da2.set_date(y, m, d);
// 调用方法set_date(),用消息y,m,d对da2的数据成员赋值
    cout << da2.get_year() << "年" << da2.get_month() << "月" << da2.get_day() <<
"日" << endl; // 调用方法输出将da2用“    年    月    日”格式输出年月日
    cout << "da2的年是否为闰年 :" << da2.isleapyear() << endl;
// 调用方法输出da2的年是否为闰年
}
```

程序设计实验

1. time.h:

```
#ifndef HTIME_H
#define HTIME_H

#include <bits/stdc++.h>
using namespace std;

class time
{
private:
    int hour;
    int minute;
    int second;
    time(int h, int m, int s): hour(h), minute(m), second(s) {}

public:
    time(): hour(0), minute(0), second(0) {}
    string output_time();
    void input_time();
    int output_hour();
    int output_minute();
    int output_second();
```

```
};

#endif
```

2. time.cpp

```
#include "htime.h"

string time::output_time()
{
    string s;
    s = to_string(hour) + ":" + to_string(minute) + ":" + to_string(second);
    return s;
}

void time::input_time()
{
    cout << "Enter hour: ";
    cin >> hour;
    cout << "Enter minute: ";
    cin >> minute;
    cout << "Enter second: ";
    cin >> second;

    if (second >= 60)
    {
        minute += second / 60;
        second %= 60;
    }
    if (minute >= 60)
    {
        hour += minute / 60;
        minute %= 60;
    }
    if (hour >= 24)
    {
        hour %= 24;
    }
}

int time::output_hour()
{
    return hour;
}

int time::output_minute()
{
    return minute;
}

int time::output_second()
```

```
{
    return second;
}
```

3. exp_104.cpp

```
#include "htime.h"

int main()
{
    class time t1;
    class time *t1_ptr = &t1;
    class time &t1_ref = t1;

    t1.input_time();
    cout << t1.output_time() << endl;
    cout << t1.output_hour() << "时" << t1.output_minute() << "分" <<
t1.output_second() << "秒" << endl;

    t1_ptr->input_time();
    cout << t1_ptr->output_time() << endl;
    cout << t1_ptr->output_hour() << "时" << t1_ptr->output_minute() << "分" <<
t1_ptr->output_second() << "秒" << endl;

    t1_ref.input_time();
    cout << t1_ref.output_time() << endl;
    cout << t1_ref.output_hour() << "时" << t1_ref.output_minute() << "分" <<
t1_ref.output_second() << "秒" << endl;
}
```