**实验五 类与对象**

**姓名：王向宇**

**专业：软件工程**

**班级：2304班**

**学号：8209230413**

**成绩：**

**【实验目的与要求】**

1、掌握声明类的方法，类和类的成员的概念以及定义对象的方法。

2、初步掌握用类和对象编制基于对象的程序。

3、学习检查和调试基于对象的程序。

**【实验内容】**

1、有以下程序：

#include<iostream>

using namespace std；

class Time // 定义Time类

{

public： // 数据成员为公用的

int hour;

int minute;

int sec ;

};

int main()

{

Time tl; //定义t1为Time类对象

cin>>t1.hour; //输入设定的时间

cin>>t1.minute;

cin>>t1.sec;

cout<<t1.hourl<<”：”<<t1.minute<<”：”<<t1.sec<<endl ;

return 0;

}

改写程序，要求：

(1)将数据成员改为私有的；

(2)将输入和输出的功能改为由成员函数实现；

(3)在类体内定义成员函数。

然后编译和运行程序。请分析什么成员应指定为公用的?什么成员应指定为私有的? 什么函数最好放在类中定义? 什么函数最好在类外定义?

2、分别给出如下的3个文件：

(1)含类定义的头文件student.h，

//student.h (这是头文件，在此文件中进行类的声明)

class Student //类声明

{

public: //公用成员函数原型声明

void display();

private:

int num;

char name[20];

char sex;

};

(2)包含成员函数定义的源文件student.cpp

//student.cpp 在此文件中进行函数的定义

#include <iostream>

#include”student.h” //不要漏写此行，否则编译通不过

void Student::display() //在类外定义display类函数

{

cout<<”num：”<<num<<endl;

cout<<”name：”<<name<<endl;

cout<<”sex：”<<sex<<endl;

}

(3)包含主函数的源文件main.cpp。

为了组成一个完整的源程序，应当有包括主函数的源文件：

//main.cpp 主函数模块

#include <iostream> //将类声明头文件包含进来

#include “student.h”

int main()

{

Student stud; //定义对象

Student stud1(007,”tcg”,’m’);

stud.display(); //执行stud对象的display函数

return 0;

}

请完善该程序，在类中增加一个对数据成员赋初值的成员函数set\_value。上机调试并运行。

3、需要求3个长方柱的体积，请编一个基于对象的程序。数据成员包括length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

(1)由键盘分别输入3个长方柱的长、宽、高；

(2)计算长方柱的体积；

(3)输出3个长方柱的体积。

请编程序，上机调试并运行。

4、建立一个对象数组，内放5个学生的（学号，成绩），设立一个函数max，用指向对象的指针作函数参数，在max函数中找出5个学生的最高成绩者，并输出其学号。

5、设有一描述坐标点的类Point，其私有变量x和y代表一个点的(x,y)坐标值。请编写程序实现以下功能：利用构造函数传递参数，在定义对象时将x、y坐标值初始化为（60,80）；利用公有成员函数void setPoint(int i, int j)将坐标值修改为(60+i,80+j)；利用公有成员函数display()输出修改后的坐标值。主函数中通过定义对象，验证各个函数。

**算法分析，程序结果**

**1.** #include<iostream>

using namespace std;

class Time // 定义Time类

{

private: // 数据成员为公用的

int hour;

int minute;

int sec;

public:

void shuru()

{

cout << "小时:";

cin >> hour; //输入设定的时间

cout << "分钟:";

cin >> minute;

cout << "秒:";

cin >> sec;

}

void shuchu()

{

cout << hour << ":" << minute << ":" << sec << endl;//输出设定的时间

}

};

int main()

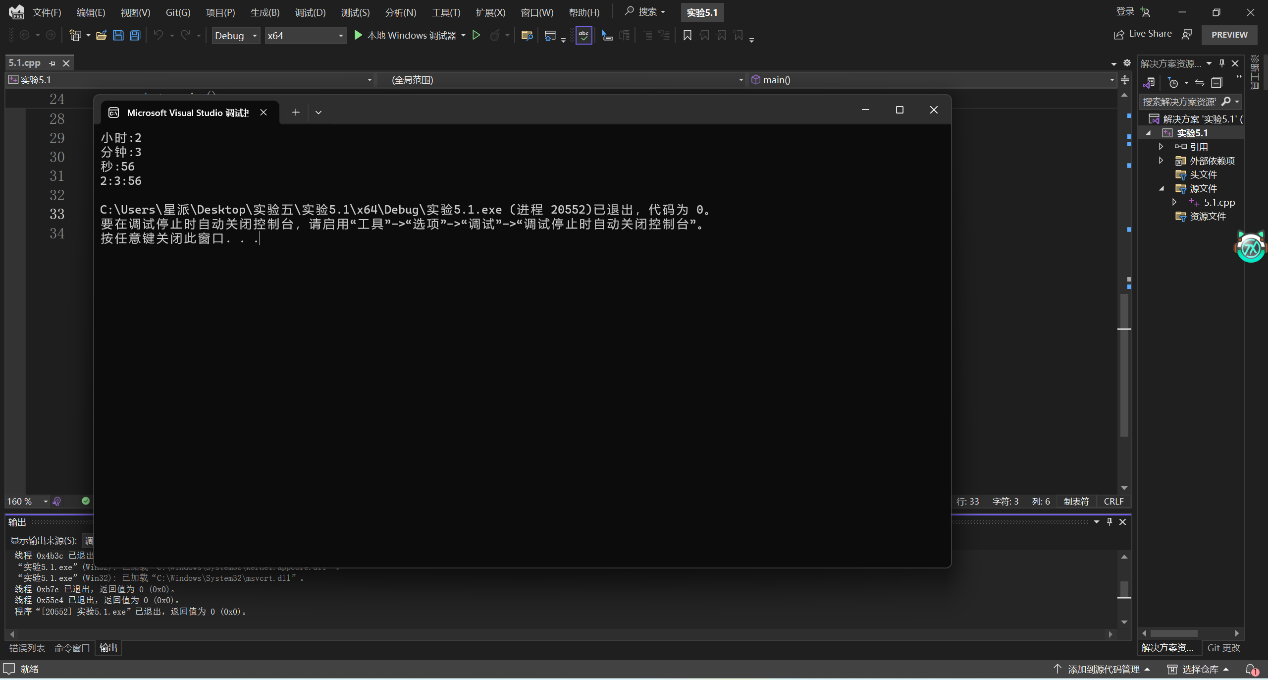
{

Time t1;

//定义t1为Time类对象

t1.shuru();

t1.shuchu();

 return 0;

}

公用的：需要在类外被直接访问的数据；成员函数

私有的：需要被保护的数据；辅助函数

类中定义的函数：访问器函数 修改器函数 构造函数和析构函数

类外定义的函数：复杂成员函数实现 友元函数 重载的操作符

2. 头文件student.h

#include <string>

class Student //类声明

{

public: //公用成员函数原型声明

void display();

void setvalue(int a,const char b [],char c)

{

num = a;

sex = c;

strcpy\_s(name, b);

}

private:

int num;

char name[20];

char sex;

};

#pragma once

源文件student.cpp

#include <iostream>

#include"student.h" //不要漏写此行，否则编译通不过

using namespace std;

void Student::display() //在类外定义display类函数

{

cout << "num：" << num << endl;

cout << "name:" << name << endl;

cout << "sex：" << sex << endl;

}

源文件main.cpp

#include <iostream> //将类声明头文件包含进来

#include "student.h"

using namespace std;

int main()

{

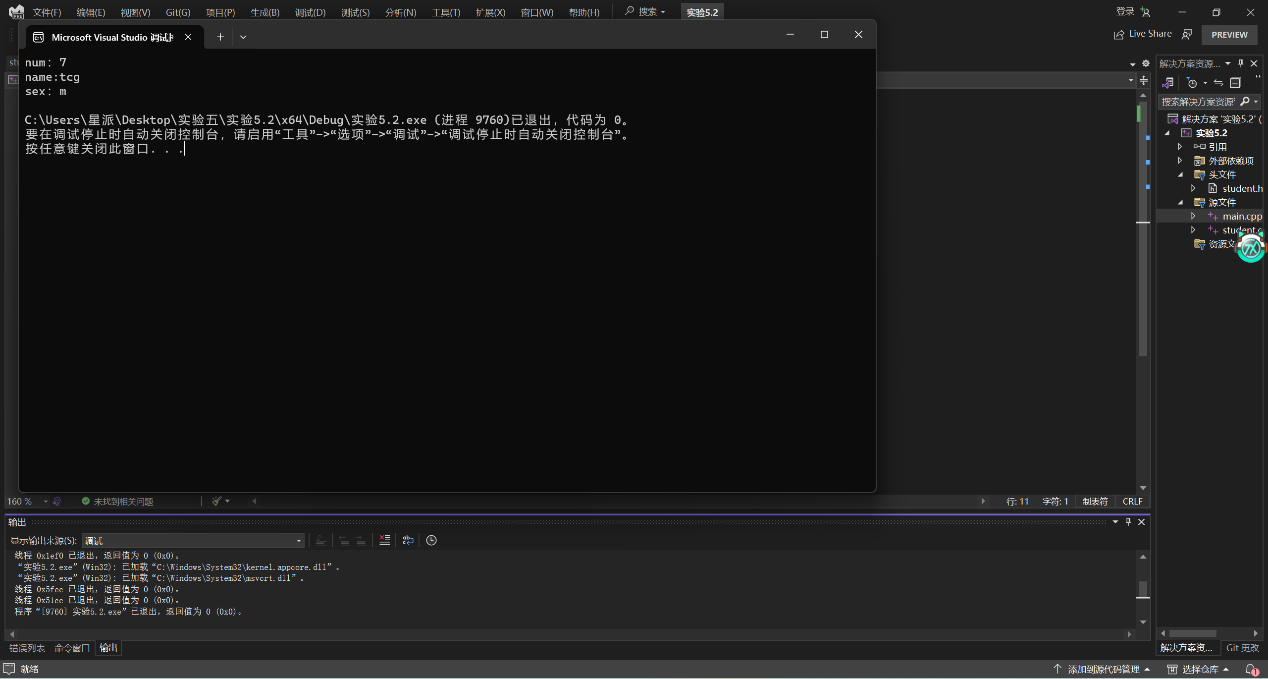
Student stud; //定义对象

stud.setvalue(007, "tcg",'m');

stud.display(); //执行stud对象的display函数

return 0;

}



3. #include<iostream>

using namespace std;

class chang

{

private: //在私有成员中定义长方体的长宽高

int length;

int width;

int height;

public:

void set()//给长方体长宽高赋值

{

cout << "长方体的长为：";

cin>>length;

cout << "长方体宽为：";

cin>>width;

cout << "长方体的高为：";

cin>>height;

}

void v()//计算并输出长方体的体积

{

cout << "长方体体积为:" << length \* width \* height << endl;

cout << endl;

}

};

int main()

{

chang a;

a.set();

a.v();

chang b;

b.set();

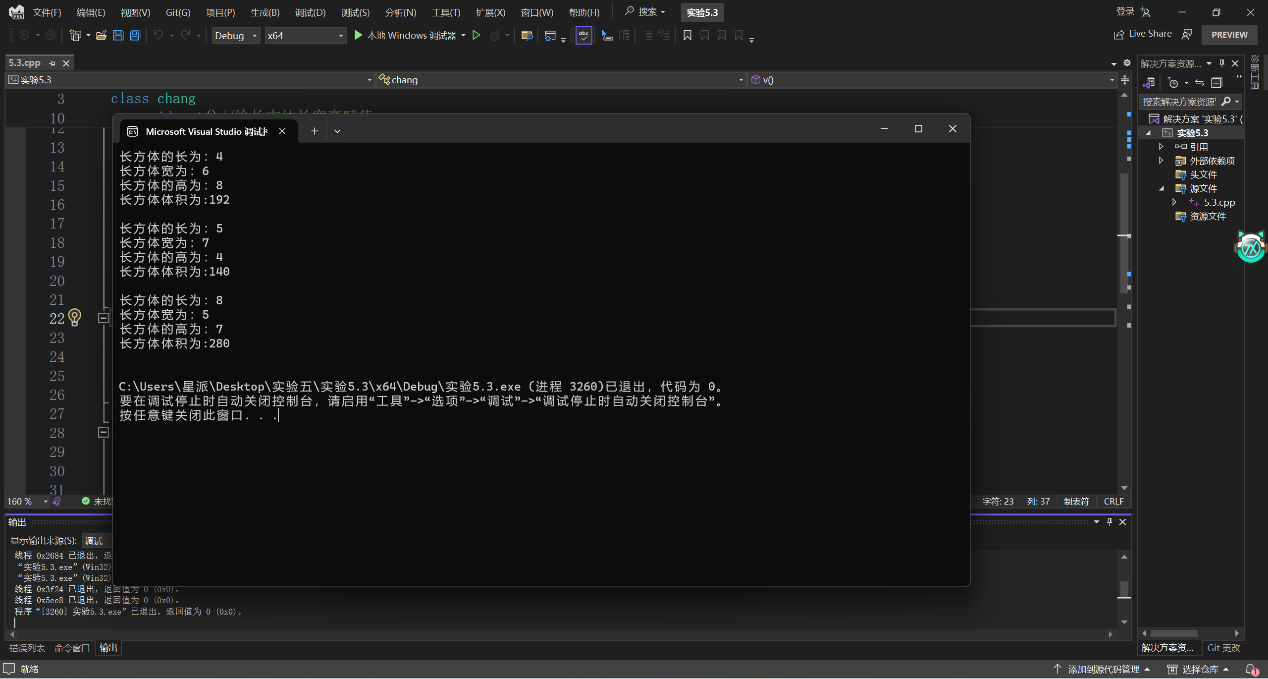
b.v();

chang c;

c.set();

c.v();

return 0;

}

4. #include<iostream>

using namespace std;

class student

{

private: //在私有成员中定义学号和得分

int ID ;

int score ;

public:

student(int a, int b) {

ID = a;//用构造函数给学号和得分赋值

score = b;

};

~student() {}//析构函数

static void max(student\* p)//传入对象指针

{

int maxscore,maxID;

for (int i = 0; i < 5; i++)

{

maxscore = p[i].score > p[i + 1].score ? p[i].score: p[i + 1].score;

}//求最高分

for (int i = 0; i < 5; i++)

{

if (maxscore == p[i].score)

{

maxID = p[i].ID;

}

}//求最高分的学号

cout << maxID << endl;//输出最高分的学号

}

};

int main()

{

student i[5] = {

student (1,5),

student (2,6),

student (3,8),

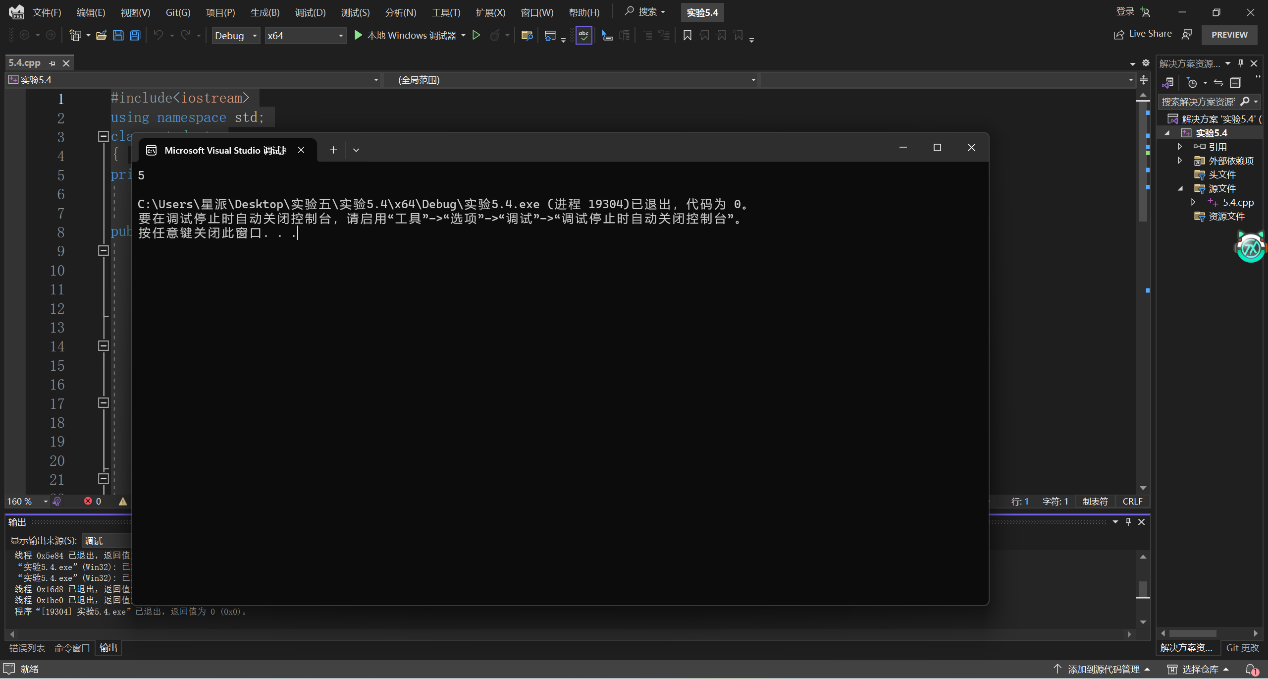
student (4,7),

student (5,9)

};//定义对象数组

student::max(i);//用静态函数并传入对象数组首地址

}



5. #include<iostream>

using namespace std;

class Point

{

private://在私有成员定义x，y

int x;

int y;

public:

Point(int a=60, int b=80)//用构造函数初始化x，y

{

x = a;

y = b;

}

~Point() {};//析构函数

void setPoint(int i, int j)//修改坐标

{

x = x + i;

y = y + j;

}

void display()//输出坐标

{

cout << "(" << x << "," << y << ")" << endl;

}

};

int main()

{

Point A;

A.setPoint(20, 30);//修改坐标

A.display();//输出坐标

}