**实验五 类与对象**

**【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。

1. 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{

private: // 数据成员为公用的

int hour;

int minute;

int sec;

public:

void showTime(){

cout << hour << ":" << minute << ":" << sec << endl;

}

void setTime(int hour,int minute,int sec){

this->hour=hour;

this->minute=minute;

this->sec=sec;

}

};

int main(){

Time t1; //定义t1为Time类对象

int hour,minute,sec;

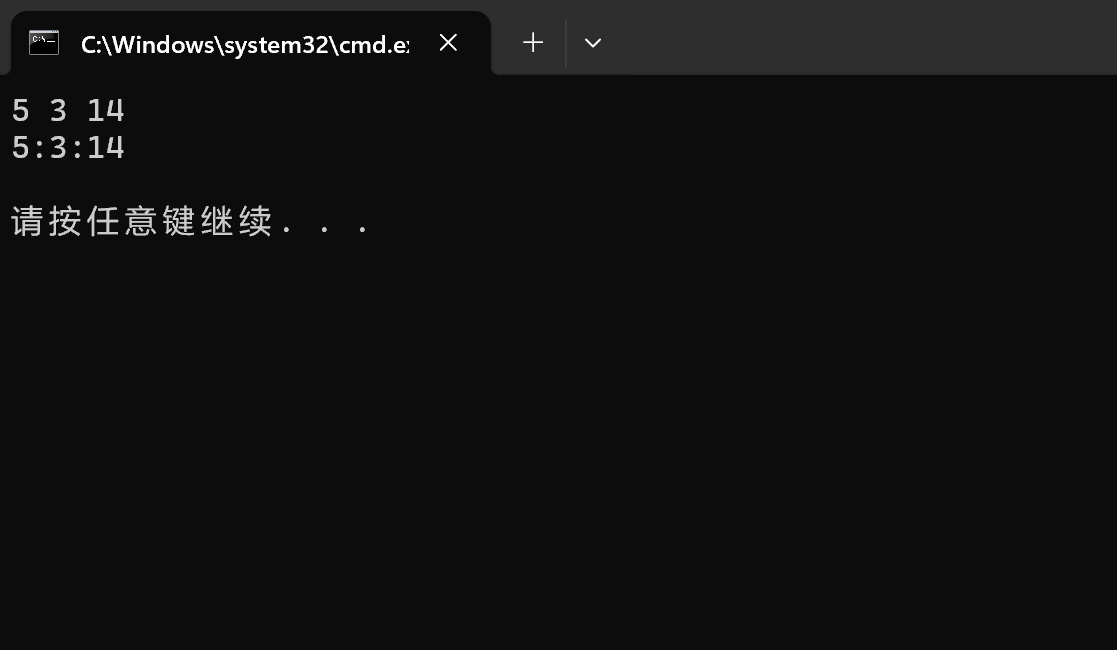
cin >> hour >> minute >> sec;

t1.setTime(hour,minute,sec);

t1.showTime();

return 0;

}

公有成员包括构造函数、析构函数以及类对外提供的接口函数。

私有成员则包括成员变量和类的内部实现细节，不应被外部直接访问。

简单且频繁调用的函数放在类内部定义。

复杂或实现细节较多的函数放在类外定义。

2、

（student.cpp）

#include "student.h"

#include <iostream>

using namespace std;

void Student::display()

{

cout << "num：" << num << endl;

cout << "name：" << name << endl;

cout << "sex：" << sex << endl;

}

void Student::set\_value(int num, const char\* name, char sex)

{

this->num = num;

for (int i = 0; i < 20; i++) {

this->name[i] = name[i];

}

this->sex = sex;

}

(student.h)

#pragma once

class Student //类声明

{

public: //公用成员函数原型声明

void display();

void set\_value(int num, const char\* name, char sex);

private:

int num;

char name[20];

char sex;

};

(main.cpp)

#include <iostream> //将类声明头文件包含进来

#include "student.h"

int main()

{

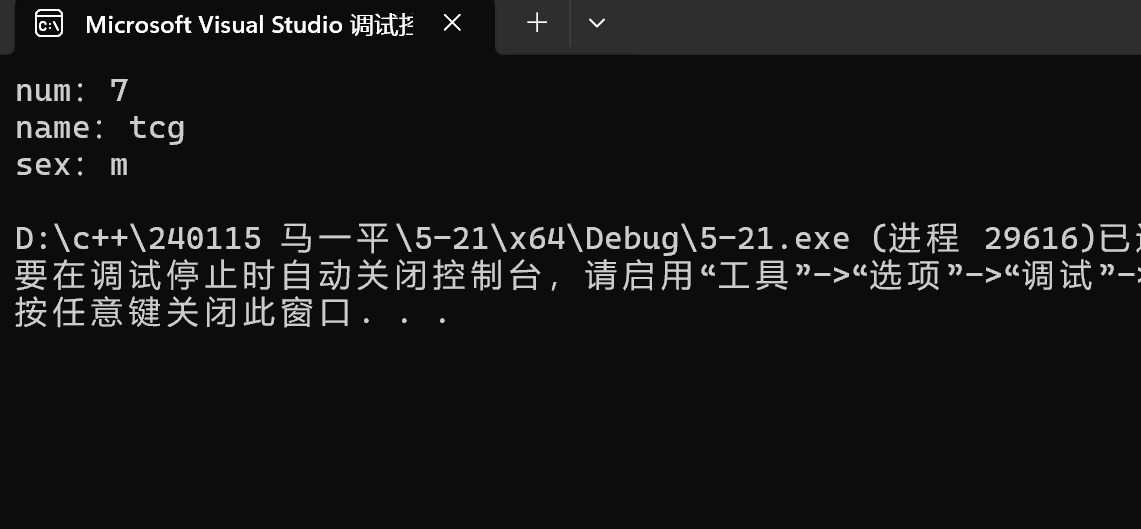
Student stud; //定义对象

stud.set\_value(7, "tcg", 'm');

stud.display(); //执行stud对象的display函数

return 0;

}



3、

#include<iostream>

using namespace std;

class zhu{

private:

int length;

int width;

int height;

public:

void V(){

cout << length\*width\*height << endl;

}

void setZhu(){

cout << "请输入长宽高：" ;

cin >> length >> width >> height;

}

};

int main(){

zhu zhu1,zhu2,zhu3;

zhu1.setZhu();

zhu1.V();

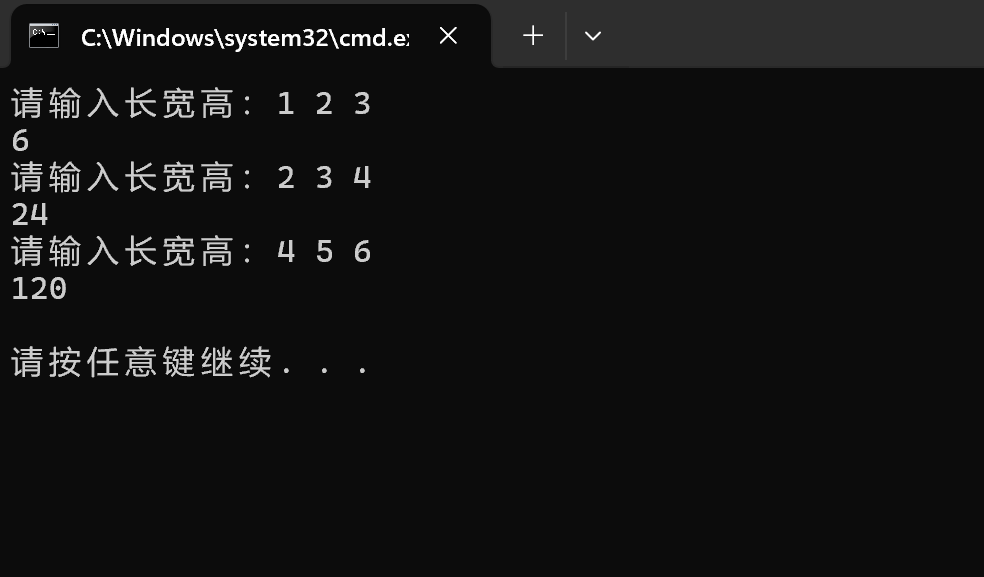
zhu2.setZhu();

zhu2.V();

zhu3.setZhu();

zhu3.V();

}

4、

#include<iostream>

using namespace std;

class Student{

public:

int num;

int score;

Student(int num,int score){

this->num=num;

this->score=score;

}

};

void max(Student\* student){

int maxi=0;

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

if(student[i].score>student[maxi].score)maxi=i;

}

cout << student[maxi].num;

}

int main(){

Student stu[5]={{1,2},{2,3},{3,4},{4,5},{5,6}};

max(stu);

}



5、

#include <iostream>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(int x = 60, int y = 80) {

this->x = x;

this->y = y;

}

void setPoint(int i, int j) {

x += i;

y += j;

}

void display() {

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

}

};

int main() {

Point p1;

p1.display();

cout << "输入i、j:";

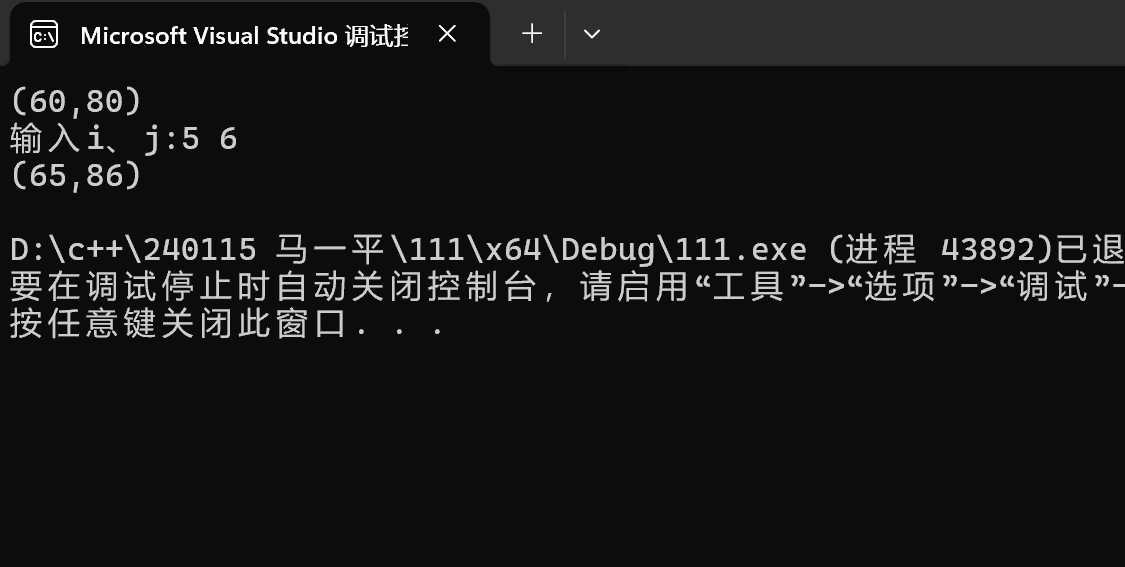
int i, j;

cin >> i >> j;

p1.setPoint(i, j);

p1.display();

}

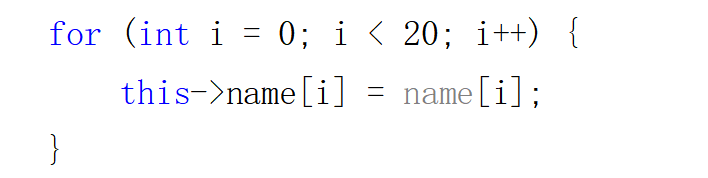


**【实验过程中遇到的问题与解决方法】**

1. **给字符串数组赋值**

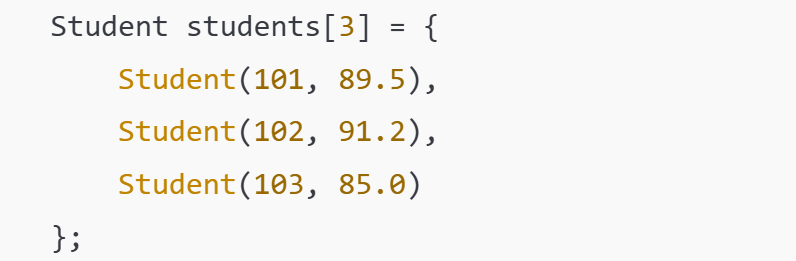
刚开始给字符串设计形参是用的类型是char \*，后来通过报错发现要用const char\*

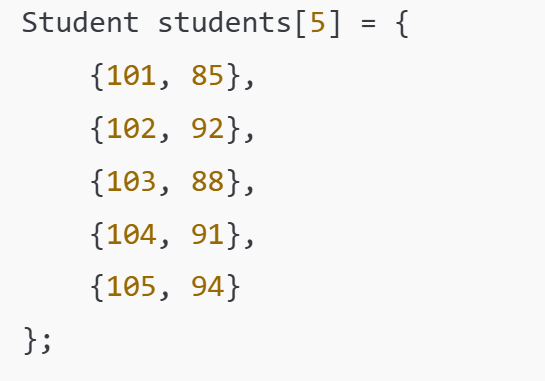
后面发现写this->name=name;时会报错，不能这样把传入的字符串赋值给字符数组，最后采用循环的方式：

成功解决了问题

1. **关于构造一个对象数组**

刚开始不知道怎么构造一个对象数组，通过上网搜索发现有两种方法





由于方法二写法更简便，故采用。

**【心得体会】**

在实验中，通过定义 Time 类、Student 类等，明确了类的构成：数据成员和成员函数。通过创建对象来实例化类，理解了对象是类的一个具体实例，每个对象有独立的状态和行为。

在将 Time 类的成员变量从公有改为私有的过程中，深刻体会到了封装的必要性。私有成员确保了数据的安全性，外部代码无法直接访问对象的内部数据，而是通过类的公有成员函数来操作数据。数据成员应当设置为私有的，以保护数据不被外部随意修改；而函数和方法应当设置为公有的，以提供对外的服务和接口。

对于类中的成员函数，我通过实验发现，较为简单的函数可以放在类体内定义，这样有助于提高代码的可读性和执行效率。比如 display() 这样的输出函数，放在类内部定义非常合适。对于复杂的函数或需要较多逻辑的函数，最好放在类外定义，以便于代码的管理和维护。