专业班级： 23级软工6班

学 号： 8209230620

姓 名： 邓入豪

**实验报告成绩：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **实验一** | **实验二** | **实验三** | **实验四** | **实验五** | **总评** |
| **成绩** |  |  |  |  |  |  |

**批阅教师：**

**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?

代码修改为：

#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 in() {

cin >> hour;

cin >> minute;

cin >> sec;

}

void show() {

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

}

};

int main()

{

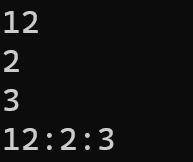
Time t1;

t1.in();

t1.show();

}

运行结果为：



问题的回答：

**用于改变类状态的成员应该定义为公有，用于保存类状态的成员应该定义为私有；**

**较为简单的函数放在类中定义；复杂的放在外部定义。**

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。

Student.h:

#pragma once

using namespace std;

//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();

void 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;

}

private:

int num;

char name[20];

char sex;

};

Student.cpp:

#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;

}

main.cpp:

#include <iostream> //将类声明头文件包含进来

#include "student.h"

int main()

{

Student stud;

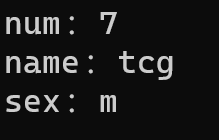
stud.set\_value(007 ,"tcg", 'm');

stud.display(); //执行stud对象的display函数

return 0;

}

运行结果为：



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。

代码为:

#include <iostream>

using namespace std;

class zhu {

private:

int length;

int height;

int width;

public:

void set\_value() {

cout << "length:";

cin >> length;

cout << "width:";

cin >> width;

cout << "height:";

cin >> height;

}

int v()

{

return (length) \* (height) \* (width);

}

};

int main() {

for (int i = 0; i < 3; i++)

{

cout << "第" << i + 1 << "个长方体" << endl;

zhu a;

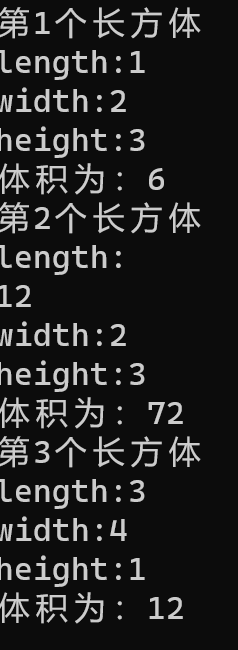
a.set\_value();

cout<<"体积为：" << a.v() << endl;

}

}

运行结果为：



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。

代码为：

#include <iostream>

using namespace std;

class student {

public:

int num;

int grades;

student(int num\_, int grades\_)

{

num = num\_;

grades = grades\_;

}

};

int max(student\* stu) {

int tem1 = 0;

int tem2 = 0;

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

{

if (((stu + i)->grades) > tem1) {

tem1 = (stu + i)->grades;

tem2 = (stu + i)->num;

}

}

return tem2;

}

int main() {

student stu[5] = {

student(1,32),

student(2,98),

student(3,100),

student(4,23),

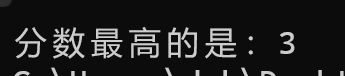
student(5,99),

};

student\* p = stu;

cout << "分数最高的是：" << max(p);

}运行结果为：



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。

代码为：

#include <iostream>

using namespace std;

class point {

private:

int x;

int y;

public:

point() {

x = 60;

y = 80;

}

void setPoint(int i, int j) {

x = 60 + i;

y = 80 + j;

}

void display() {

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

}

};

int main() {

point p1;

point p2;

cout << "p1:";

p1.display();

cout << "p2:";

p2.display();

p2.setPoint(121,312);

p2.display();

}

运行结果为：