# 第五次实验报告

# 题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 input(Time &input\_t)//输入对象的属性

{

Time temp;

cout << "输入hour:" << '\t';

cin >> temp.hour;

if (temp.hour > 23 || temp.hour < 0)

{

cout << "输入错误" << endl;

system("pause");

}

cout << "输入minute:" << '\t';

cin >> temp.minute;

if (temp.minute < 0 || temp.minute>60)

{

cout << "输入错误" << endl;

system("pause");

}

cout << "输入sec:" << '\t';

cin >> temp.sec;

if (temp.sec < 0 || temp.sec>60)

{

cout << "输入错误" << endl;

system("pause");

}

input\_t = temp;

}

void output(Time output\_t)//输出对象

{

cout << output\_t.hour <<":";

cout << output\_t.minute << ":";

cout << output\_t.sec << endl;

}

};

int main()

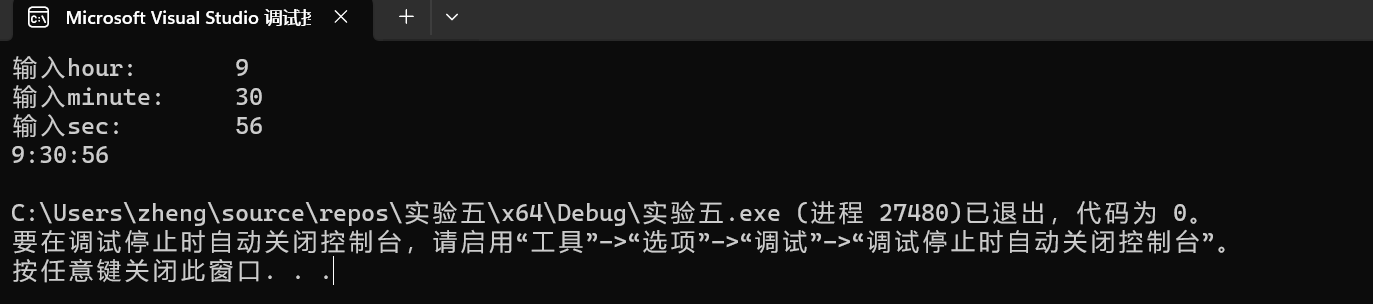
{

Time tl; //定义t1为Time类对象

tl.input(tl);

tl.output(tl);

return 0;

}

# 题2：

Student.h:

#pragma once

//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( Student\* const p);

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

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;

}

void Student::set\_value(Student\* const p)

{

cout << "请输入num：" << '\n';

cin >> p->num;

cout << "请输入name：" << '\n';

cin >> p->name;

cout << "请输入sex(男用m表示，女用f表示）：" << '\n';

cin >> p->sex;

if (p->sex != 'm' && p->sex != 'f')

{

cout << "输入性别错误" << endl;

}

}

**Main.cpp:**

#include<iostream>

#include"student.h"

using namespace std;

int main()

{

//main.cpp 主函数模块

Student stud; //定义对象

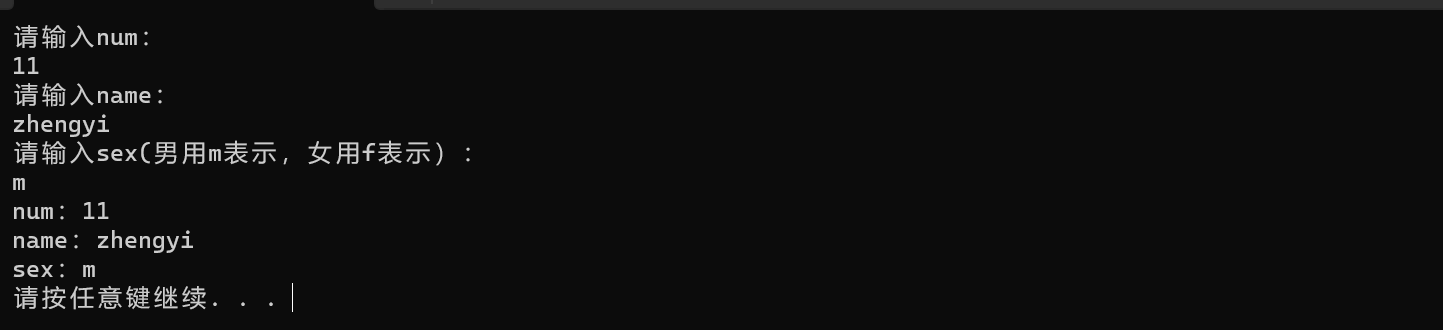
//Student stud1(007, "tcg", 'm');

Student\* p = &stud;

stud.set\_value(p);

stud.display(); //执行stud对象的display函数

system("pause");

}****

# 题3：

#include<iostream>

using namespace std;

class Cuboid

{

private:

int length;

int width;

int high;

int V;

public:

void setvalue(Cuboid\*p)

{

cout << "输入长方体的长：";

cin >> p->length;

cout << "输入长方体的宽：";

cin >> p->width;

cout << "输入长方体的高：";

cin >> p->high;

}

void calculate(Cuboid\*p)

{

p->V = p->length \* p->width \* p->high;

}

void display(Cuboid temp)

{

cout << temp.V << endl;;

}

};

int main()

{

//定义

Cuboid a,b,c;

Cuboid\* pa = &a;

Cuboid\* pb = &b;

Cuboid\* pc = &c;

//赋值

cout << "输入第一个长方体的属性" << endl;

a.setvalue(pa);

cout << "输入第二个长方体的属性" << endl;

a.setvalue(pb);

cout << "输入第三个长方体的属性" << endl;

a.setvalue(pc);

//运算

a.calculate(pa);

a.calculate(pb);

a.calculate(pc);

//输出

cout << "第一个长方体的体积为：";

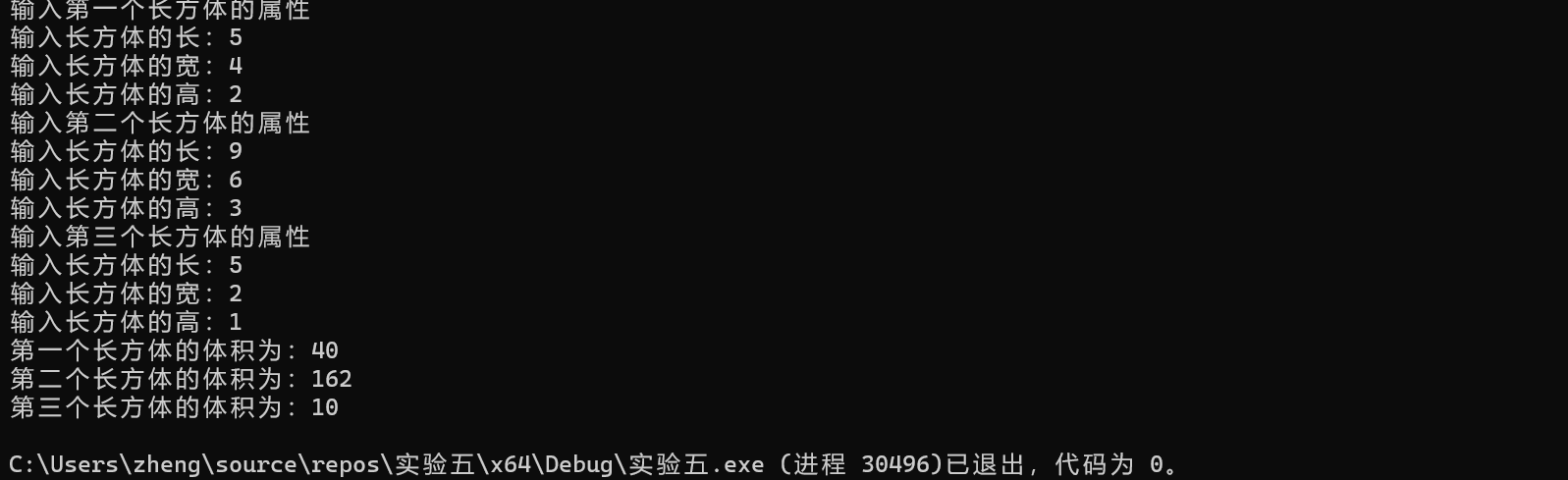
a.display(a);

cout << "第二个长方体的体积为：";

a.display(b);

cout << "第三个长方体的体积为：";

a.display(c);

}****

# 题4

#include<iostream>

using namespace std;

class student

{

private:

int id;

int score;

friend class group;

};

class group

{

private:

student arry[5];

public:

//输入对象数组的属性

void setvalue(group\*pa)

{

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

{

cout << "输入第" << i + 1 << "个学生的id：";

cin >> pa->arry[i].id;

cout << "输入第" << i + 1 << "个学生的成绩：";

cin >> pa->arry[i].score;

}

}

//选择数组中分数高的，输出学号

void max(group\*pa)

{

for (int i = 0; i < 4; i++)

{

for (int j = i + 1; j < 5; j++)

{

if (pa->arry[i].score < pa->arry[j].score)

{

student temp;

temp = pa->arry[i];

pa->arry[i] = pa->arry[j];

pa->arry[j] = temp;

}

}

}

cout << "成绩最高的学生id为：";

cout << pa->arry[0].id << endl;

}

};

int main()

{

group aa;

group \*pa = &aa;

aa.setvalue(pa);

aa.max(pa);

system("pause");

}

# 题5：

#include<iostream>

using namespace std;

class point

{

private:

int x;

int y;

public:

//构造函数

point(int i=60,int j=80)

{

x = i;

y = j;

}

void setPoint(int i, int j)//将坐标值修改为(60+i,80+j)

{

x = x + i;

y = y + j;

}

void display(point a)

{

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

}

};

int main()

{

int i, j;

point a;

cout << "输入i=";

cin >> i;

cout << "输入j=";

cin >> j;

a.setPoint(i, j);

a.display(a);

system("pause");

}

# 遇到的问题与解决方法：

这次的实验相对与前两次对逻辑的思维能力并没有那么高，主要是对近期学过的类与对象相关知识的运用。主要的难点也在于对知识的熟练程度。在编程的过程中，有时会对一些细节性的知识有遗漏，导致代码出现bug。这时主要的解决方法就是针对相关知识点进行复习，往往把知识点搞懂后，问题也就迎刃而解了。

# 体会：

这次试验最大的收获在于对知识点的熟练掌握与运用，不只是这一次相关的知识。更重要的还是学习知识的方法，对于知识不能一知半解，而是要及时理解。这样才能对其更加熟练的运用。