**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。

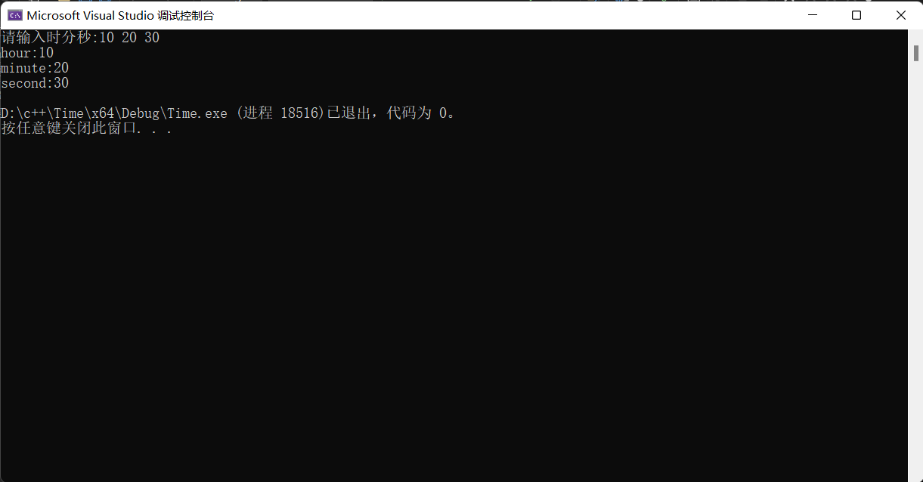
5、设有一描述坐标点的类Point，其私有变量x和y代表一个点的(x,y)坐标值。请编写程序实现以下功能：利用构造函数传递参数，在定义对象时将x、y坐标值初始化为（60,80）；利用公有成员函数void setPoint(int i, int j)将坐标值修改为(60+i,80+j)；利用公有成员函数display()输出修改后的坐标值。主函数中通过定义对象，验证各个函数。

【算法分析、程序结果】

第一题实验代码:

1. #include <iostream>
2. using namespace std;
3. class Time {
4. int hour;
5. int minute;
6. int sec;
7. public:
8. void Set();
9. void Show();
10. };
11. void Time::Set() {
12. int h, m, s;
13. cout<<"请输入时分秒:";
14. cin >> h >> m >> s;
15. if(h>=24||h<0||m<0||m>=60||s<0||s>=60){
16. cout<<"这不是一个正确的时间,请重新输入:"<<"\n";
17. Set();
18. return;
19. }
20. else{
21. hour = h;
22. minute = m;
23. sec = s;
24. }
26. }
27. void Time::Show() {
28. cout <<"hour:"<< hour << "\n";
29. cout <<"minute:"<< minute << "\n";
30. cout <<"second:"<< sec << "\n";
31. }
32. int main() {
33. Time tl;
34. tl.Set();
35. tl.Show();
36. return 0;
37. }

第一题实验结果:



第二题实验代码:

student.h文件

1. *//student.h*
2. class Student {
3. public:
4. Student(); *//默认构造函数*
5. Student(int num, const char\* name, char sex); *//重载构造函数*
6. void set\_value(int num, const char\* name, char sex); *//数据成员赋初值的成员函数*
7. void display();
8. private:
9. int num;
10. char name[20];
11. char sex;
12. };

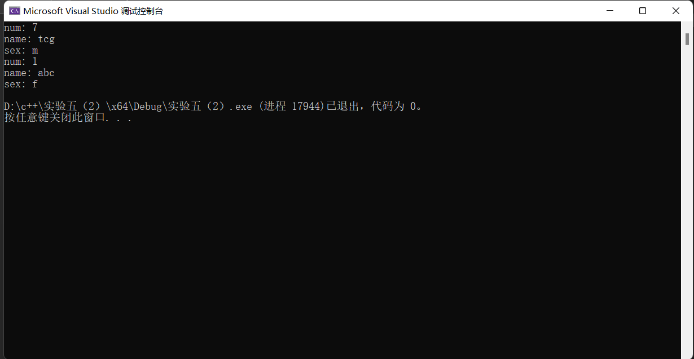
student.cpp文件

1. *//student.cpp*
2. #include <iostream>
3. #include "student.h"
4. Student::Student() { *//默认构造函数的定义*
5. num = 0;
6. name[0] = '\0';
7. sex = '\0';
8. }
9. Student::Student(int num, const char\* name, char sex) { *//重载构造函数的定义*
10. this->num = num;
11. strcpy\_s(this->name, name);
12. this->sex = sex;
13. }
14. void Student::set\_value(int num, const char\* name, char sex) { *//数据成员赋初值的成员函数的定义*
15. this->num = num;
16. strcpy\_s(this->name, name);
17. this->sex = sex;
18. }
19. void Student::display() { *//display函数的定义*
20. std::cout << "num: " << num << std::endl;
21. std::cout << "name: " << name << std::endl;
22. std::cout << "sex: " << sex << std::endl;
23. }

main.cpp文件

1. *//main.cpp*
2. #include <iostream>
3. #include "student.h"
4. int main() {
5. Student stud1;
6. stud1.set\_value(007, "tcg", 'm'); *//使用set\_value函数设置对象的数据成员的值*
7. stud1.display(); *//调用display函数显示对象的数据成员的值*
8. Student stud2(001, "abc", 'f'); *//使用重载构造函数创建另一个对象*
9. stud2.display(); *//调用display函数显示对象的数据成员的值*
10. return 0;
11. }

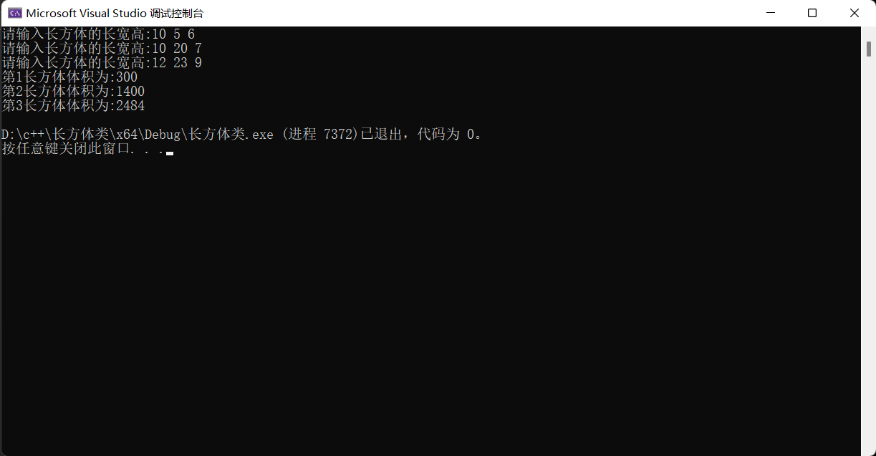
第二题实验结果:



第三题实验代码:

1. #include <iostream>
2. using namespace std;
3. class cuboid {
4. int length;
5. int width;
6. int height;
7. public:
8. void Set(int length, int width, int height);
9. int count\_volume();
10. void Show\_volume();
11. };
12. void cuboid::Set(int length, int width, int height) {
13. this->length = length;
14. this->width = width;
15. this->height = height;
16. }
17. int cuboid::count\_volume() {
18. int volume = length \* width \* height;
19. return volume;
20. }
21. void cuboid::Show\_volume() {
22. cout << "长方体体积为:" << count\_volume() << "\n";
23. }
24. int main() {
25. const int count = 3;
26. cuboid cuboid[count];
27. for (int i = 0; i < 3; i++) {
28. int length, width, height;
29. cout << "请输入长方体的长宽高:";
30. cin >> length >> width >> height;
31. cuboid[i].Set(length, width, height);
32. }
33. for (int i = 0; i < 3; i++) {
34. cout<<"第"<<(i+1);
35. cuboid[i].Show\_volume();
36. }
37. return 0;
38. }

第三题实验结果:



第四题实验代码:

1. #include <iostream>
2. using namespace std;
3. class student {
4. int id;
5. int score;
6. public:
7. void Set(int id, int score);
8. int Get\_score();
9. int Get\_id();
10. };
11. void student::Set(int id, int score) {
12. this->id = id;
13. this->score = score;
14. }
15. int student::Get\_score() {
16. return score;
17. }
18. int student::Get\_id() {
19. return id;
20. }
21. student\* max(student\* Student, int num) {
22. student\* max\_student = &Student[0];
23. for (int i = 0; i < num; i++) {
24. if (Student[i].Get\_score() > max\_student->Get\_score()) {
25. max\_student = &Student[i];
26. }
27. }
28. return max\_student;
29. }
30. int main() {
31. const int count = 5;
32. int id, score;
33. student Student[count];
34. for (int i = 0; i < count; i++) {
35. cout << "请输入学生的学号和成绩:";
36. cin >> id >> score;
37. Student[i].Set(id, score);
38. }
39. student\* highest\_student = max(Student, count);
40. cout << "最高成绩学生的学号为:" << highest\_student->Get\_id() << "\n";
41. return 0;
42. }

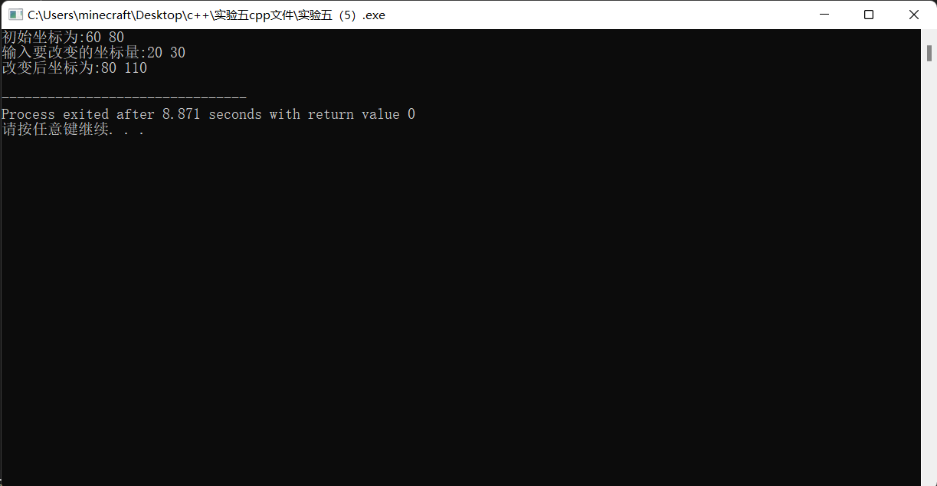
第四题实验结果:



第五题实验代码:

1. #include <iostream>
2. using namespace std;
3. class point {
4. int x;
5. int y;
6. public:
7. void setPoint(int i, int j);
8. void display();
9. point();
10. point(int x, int y);
11. };
12. point::point() {
13. x = 60;
14. y = 80;
15. }
16. point::point(int x, int y) {
17. this->x = x;
18. this->y = y;
19. }
20. void point::setPoint(int i, int j) {
21. x = x + i;
22. y = y + j;
23. }
24. void point::display() {
25. cout << x <<" "<< y << "\n";
26. }
27. int main() {
28. point Point;
29. cout << "初始坐标为:";
30. Point.display();
31. int i, j;
32. cout << "输入要改变的坐标量:";
33. cin >> i >> j;
34. Point.setPoint(i, j);
35. cout << "改变后坐标为:";
36. Point.display();
37. return 0;
38. }

第五题实验结果:



【遇到的问题和解决办法】

在本次实验中，主要解决了一些与类相关的问题，包括什么是类，类的公有成员和私有成员，以及构造函数等等的问题，在解决问题的过程中没有遇到太多的问题，不过在解决有关类的构造函数时，开始并不能熟练的使用相关的方法来解决问题，在经过细致的复习后，便熟悉的掌握了相关的方法，此外还有在第四个问题中，有关指向对象的指针开始时也不太熟悉，后来与数组指针以及一般的指针类比后也掌握相关的方法，实现用指向对象的指针作为函数参数来传递参数来完成相关的功能。

【体会】

通过这次实验，我掌握了基础的与类相关的知识和具体的代码实现，包括类的定义，类的公有和私有成员，知道了数据相关的有关成员应该设为私有，应当通过公有的成员函数来调用和更改类的私有成员，了解了使用成员函数这一实现类相关功能的重要方式，这样能使代码更加模块化和易于管理，此外还了解了构造函数相关的知识，了解了怎么给一个函数中私有的数据成员赋以初始值，总之，在此次实验中，初步了解了类相关的知识，并且能将知识运用在具体的内容中，提升了编写代码的能力。

最后，本次实验只是简单的学习了部分与类相关的知识，还有着更多更全面的知识需要去学习，希望在之后的实验也能很好的完成。