**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软件工程2402班

学 号： 8209240214

姓 名： 程世勋

**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?

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。

程序

1.

#include <iostream>

using namespace std;

class Time // 定义Time类

{

private: // 数据成员为私有的

int hour;

int minute;

int sec;

public: // 公有成员函数

void inputTime() // 输入时间的成员函数

{

cout << "请输入时间（小时 分钟 秒）: ";

cin >> hour >> minute >> sec;

}

void displayTime() // 输出时间的成员函数

{

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

}

};

int main()

{

Time t1; // 定义t1为Time类对象

t1.inputTime(); // 调用成员函数输入时间

t1.displayTime(); // 调用成员函数输出时间

return 0;

}

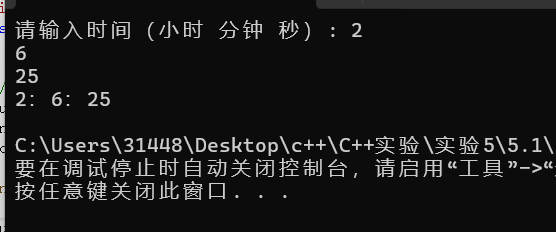
（3）

私有成员: 数据成员 hour, minute, sec 被设置为私有是为了封装数据，防止外部直接访问和修改这些值。这样可以确保数据的有效性和一致性。

公用成员: 成员函数 inputTime() 和 displayTime() 被设置为公用，以便外部可以调用这些函数来输入和输出时间。这样做可以提供对私有数据的受控访问。

最好放在类中定义的函数: inputTime() 和 displayTime() 这类函数通常与类的内部状态密切相关，且实现相对简单，适合放在类内定义。

最好在类外定义的函数: 如果函数的实现较复杂或需要较多的逻辑处理，通常可以在类外定义，以便保持类定义的简洁性。例如，涉及复杂的算法或者多个类交互的函数可以放在类外。



2.

//student.h

#define STUDENT\_H

class Student // 类声明

{

public: // 公用成员函数原型声明

void display(); // 显示学生信息

void set\_value(int n, const char\* nm, char s); // 设置学生信息

private:

int num; // 学号

char name[20]; // 姓名

char sex; // 性别

};

//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::set\_value(int n, const char\* nm, char s) // 定义set\_value函数

{

num = n; // 设置学号

strncpy\_s(name, nm, sizeof(name) - 1); // 设置姓名

name[sizeof(name) - 1] = '\0'; // 确保字符串以'\0'结尾

sex = s; // 设置性别

}

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; // 定义对象

stud.set\_value(007, "tcg", 'm'); // 设置学生信息

stud.display(); // 执行stud对象的display函数

return 0;

}



3.

// Box.h

#define BOX\_H

class Box

{

public:

void setDimensions(double l, double w, double h); // 设置长方柱的长、宽、高

double getVolume() const; // 计算体积

void displayVolume() const; // 输出体积

private:

double length; // 长

double width; // 宽

double height; // 高

};

// Box.cpp

#include <iostream>

#include "Box.h"

using namespace std;

void Box::setDimensions(double l, double w, double h)

{

length = l;

width = w;

height = h;

}

double Box::getVolume() const

{

return length \* width \* height; // 体积计算公式：长 × 宽 × 高

}

void Box::displayVolume() const

{

cout << getVolume() << endl; // 输出体积

}

// main.cpp

#include <iostream>

#include "Box.h"

using namespace std;

int main()

{

Box box1, box2, box3; // 创建3个长方柱对象

double length, width, height;

cout << " 输入第一个长方柱的长、宽、高 :";

cin >> length >> width >> height;

box1.setDimensions(length, width, height);

cout << "输入第二个长方柱的长、宽、高 :";

cin >> length >> width >> height;

box2.setDimensions(length, width, height);

cout << "输入第三个长方柱的长、宽、高: ";

cin >> length >> width >> height;

box3.setDimensions(length, width, height);

// 输出3个长方柱的体积

cout << "3个长方柱的体积依次是：" << endl;

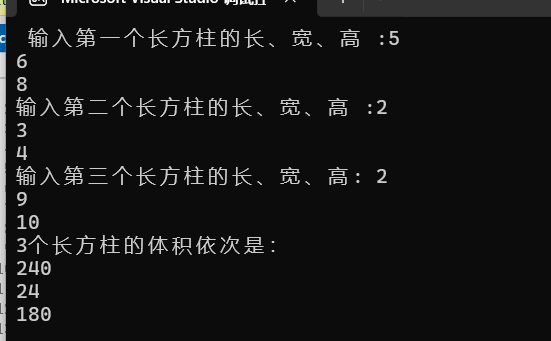
box1.displayVolume();

box2.displayVolume();

box3.displayVolume();

return 0;

}



4.

#include <iostream>

using namespace std;

// Student 类定义

class Student

{

public:

void setData(int id, double score); // 设置学生的学号和成绩

int getId() const; // 获取学生的学号

double getScore() const; // 获取学生的成绩

private:

int id; // 学号

double score; // 成绩

};

// 实现 Student 类的成员函数

void Student::setData(int id, double score)

{

this->id = id;

this->score = score;

}

int Student::getId() const

{

return id;

}

double Student::getScore() const

{

return score;

}

void max(Student\* students, int n) // max 函数，用指向学生对象的指针作为参数，找出成绩最高的学生

{

Student\* maxStudent = &students[0]; // 假设第一个学生是成绩最高的

for (int i = 1; i < n; ++i)

{

if (students[i].getScore() > maxStudent->getScore())

{

maxStudent = &students[i]; // 找到成绩更高的学生

}

}

cout << "成绩最高的学生是: " << maxStudent->getId() << endl;

}

int main()

{

Student students[5]; // 创建一个包含5个学生的数组

int id;

double score;

for (int i = 0; i < 5; ++i) // 输入5个学生的学号和成绩

{

cout << "输入5个学生的学号和成绩： " << i + 1 << ": ";

cin >> id >> score;

students[i].setData(id, score);

}

// 找出成绩最高的学生并输出其学号

max(students, 5);

return 0;

}



5.

#include <iostream>

using namespace std;

class Point

{

private:

int x, y; // 私有成员变量 x 和 y，表示坐标

public:

Point() // 构造函数，初始化坐标为 (60, 80)

{

x = 60;

y = 80;

}

void setPoint(int i, int j) // 公有成员函数，修改坐标值

{

x = 60 + i;

y = 80 + j;

}

void display() const // 公有成员函数，显示当前坐标

{

cout << "当前坐标为: (" << x << ", " << y << ")" << endl;

}

};

int main()

{

Point p1; // 创建 Point 类的对象，自动调用构造函数，将坐标初始化为 (60, 80)

cout << "初始坐标为： "; // 输出初始坐标

p1.display();

int i, j;// 用户输入要修改的坐标偏移量

cout << "输入要修改的x坐标偏移量: ";

cin >> i;

cout << "输入要修改的y坐标偏移量: ";

cin >> j;

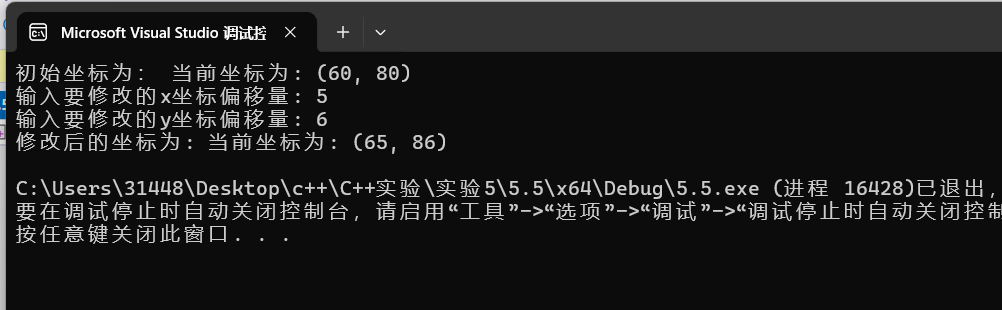
p1.setPoint(i, j); // 修改坐标值，调用 setPoint 函数

cout << "修改后的坐标为: ";

p1.display();

return 0;

}



1. **遇到的问题与解决方法**

**1.**

问题：不了解this指针的用法，不知道怎么用指向对象的指针作函数参数。

解决方法：

阅读书籍，了解this指针的用法，并学习用指向对象的指针作函数参数的相关知识。

void max(Student\* students, int n) // max 函数，用指向学生对象的指针作为参数，找出成绩最高的学生

**2**.问题： 实验5不知道怎么让用户输入要修改的坐标偏移量，只会写成一条语句：

p1.setPoint(10, 20);

解决方法：

看书并上网查阅资料，发现可以定义变量i和j由用户输入，再写成如下语句，即可实现改功能。

p1.setPoint(i, j); // 修改坐标值，调用 setPoint 函数

**四、体会**

这次实验内容较少，整体难度不大，做完所花的时间不多，一方面体现了我对c++的理解得更加深入，另一方面说明了我敲代码的速度变得更快。

通过这次实验，我初步掌握了用类和对象编制基于对象的程序，知道了将程序分为.h ,main.cpp , .cpp的方式来编写代码，以此使代码的编写变得更加规范和便于理解。尤其是用指向对象的指针作函数参数的this指针，我觉得是我学习c++过程中的难点，但经过我的努力学习，也成功攻克了这个难关。

综上，这次实验使我意识到许多c++的难点我们都可以慢慢地解决，就像类和对象的概念一样，是一个自底向上的过程。