**计算机程序设计基础（C++）**

**实验报告**

专业班级 软件工程2301班

学 号 8209230104

姓 名 向一洋

**实验报告成绩：**

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

**批阅老师:\_\_\_\_\_\_\_\_**

**实验五 类与对象**

**一、实验目的**

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。

**二、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;

};

1. 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;

}

1. 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、编写程序：定义抽象基类 Shape，由它派生出五个派生类：Circle（圆形）、Square（正方形）、Rectangle（ 长方形）、Trapezoid （梯形）和 Triangle （三角形），用虚函数分别计算各种图形的面积，并求出它们的和。要求用基类指针数组。使它的每一个元素指向一个派生类的对象。

注：主函数中定义如下对象

Circle circle(10.2); Square square(3); Rectangle rectangle(3,4);

Trapezoid trapezoid(2.0,4.5,3); Triangle triangle(4,5,6);

5、设计一个函数，返回两个向量的交集，函数头如下： template<typename T>

vector<T> intersection(const vector<T>& v1, const vector<T>& v2)

两个向量的交际为它们的共同包含的元素，编写一个测试程序，提示用户输入两个向量，每个包含 5 个字符串，打印输出它们的交集。

**三、实验步骤、算法和结果分析**

1.思考：一般函数成员指定为公有的，而数据成员指定为私有的。如果一个函数2-3行，则在类内定义，如果函数多于3行，则在类内声明，类外定义。

程序：

#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 setTime()

{

int x, y, z;

cin >> x >> y >> z;

hour = x;

minute = y;

sec = z;

}

int getTime()

{

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

return 0;

}

};

int main()

{

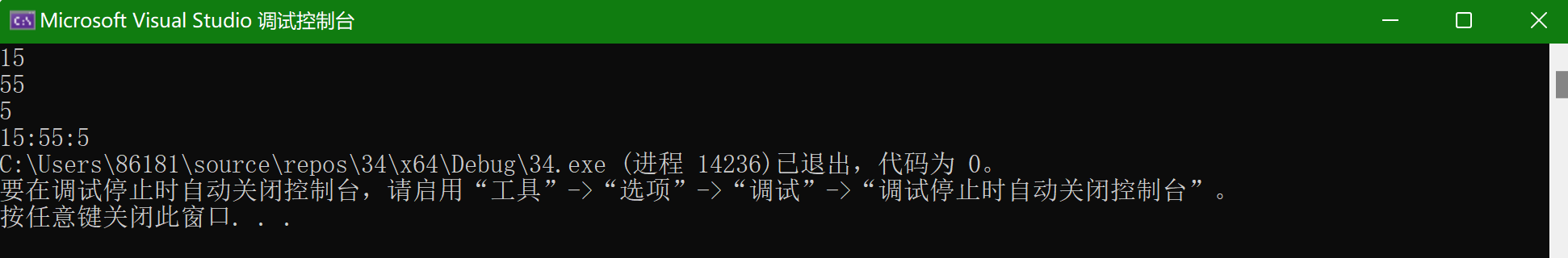
Time tl; //定义 t1 为 Time 类对象

tl.setTime();

tl.getTime();

return 0;

}

结果：

2.

程序：

//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

stud.set\_value();

stud.display(); //执行 stud 对象的 display 函数

return 0;

}

//student.cpp 在此文件中进行函数的定义

#include <iostream>

using namespace std;

#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;

}

void Student::set\_value()

{

cout << "The age of the student is:";

cin >> num;

cout << "The name of the student is:";

cin >> name;

cout << "The sex of the student is:";

cin >> sex;

}

//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();

private:

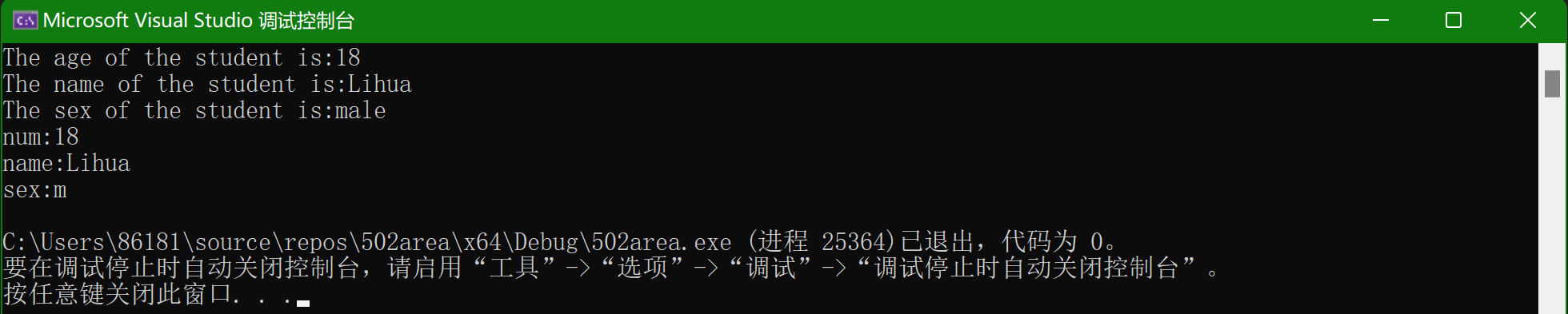
int num;

char name[20];

char sex;

};

结果：



创建对象->三个长方柱

3.算法：

c1,c2,c3

长方柱类

数据成员

成员函数

Input

赋值函数set\_value()

体积函数get\_area()

Output

程序：

#include<iostream>

using namespace std;

class Column

{

private:

int length, width, height;

public:

void set\_value();//赋值函数的声明~

int get\_area();//求面积函数的声明~

};

void Column::set\_value()

{

cout << "length、width、height：";

cin >> length >> width >> height;

}//给长方柱赋值~

int Column::get\_area()

{

int area = length \* width \* height;

return area;

}//定义求面积函数~

int main()

{

Column c1;

c1.set\_value();

cout << c1.get\_area() << endl;

Column c2;

c2.set\_value();

cout << c2.get\_area() << endl;

Column c3;

c3.set\_value();

cout << c3.get\_area() << endl;

system("pause");

return 0;

}

结果：



4.算法：

Circle(10.2)

Shape类

Square(3)

成员函数

Rectangle(3,4)

纯虚area()

求面积

Trapezoid(2.0,4.5,3)

输出area

Triangle(4,5,6)

Add

程序：

#include<iostream>

#include <cmath>//求根号要包括

using namespace std;

//创建Shape基类

class Shape

{

public:

//定义求面积的纯虚函数

virtual double area() = 0;

};

//创建子类Circle圆形

class Circle :public Shape

{

public:

//圆形求面积：Π\*r\*r

virtual double area()

{

return NP \* r \* r;

}

Circle(double R)

{

r = R;

cout<< "圆形的面积为：" <<area() << endl;

}

private:

double r;

const double NP = 3.14;

};

//创建子类Square正方形

class Square :public Shape

{

public:

//正方形求面积：l\*l

virtual double area()

{

return l \* l;

}

Square(double L)

{

l = L;

cout << "正方形的面积为：" << area() << endl;

}

private:

double l;//正方形的边长

};

//创建子类长方形Rectangle

class Rectangle :public Shape

{

public:

//长方形面积：a\*b

virtual double area()

{

return a \* b;

}

Rectangle(double A, double B)

{

a = A;

b = B;

cout << "长方形的面积为：" << area() << endl;

}

private:

double a; //长

double b;//宽

};

//创建子类梯形Trapezoid

class Trapezoid :public Shape

{

public:

//梯形面积：(up+down)\*h/2

virtual double area()

{

return (up + down) \* h / 2;

}

Trapezoid(double UP, double Down, double H)

{

up = UP;

down = Down;

h = H;

cout << "梯形的面积为：" << area() << endl;

}

private:

double up;//上底

double down;//下底

double h;//高

};

//创建子类三角形Triangle

class Triangle :public Shape

{

public:

//求三角形面积：海伦公式

virtual double area()

{

double p = (a1 + b1 + c1) / 2;

double S=0.0;//面积

S =sqrt( p \* (p - a1) \* (p - b1) \* (p - c1));//求根号用sqrt（）

return S;

}

Triangle(double A1, double B1, double C1)

{

a1 = A1;

b1 = B1;

c1 = C1;

cout << "三角形的面积为：" << area() << endl;

}

private:

double a1, b1, c1;//三角形三边

};

int main()

{//创建子类对象

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3, 4);

Trapezoid trapezoid(2.0, 4.5, 3);

Triangle triangle(4, 5, 6);

//基类指针数组每一个元素指向一个派生类的对象

Shape\* shape[5] = { &circle, &square, &rectangle, &trapezoid, &triangle };

//接下来求和

double sum = 0.0;

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

{

sum += shape[i]->area();

}

cout << "面积和为：" << sum << endl;

system("pause");

return 0;

}

结果：



5.

程序：

#include<iostream>

#include<vector> //引用vertor的头文件

#include<string>

using namespace std;

template<typename T>

vector<T>intersection(const vector<T>& v1, const vector<T>& v2) //定义函数intersection

{

vector<T> v3;

for (int i = 0; i < v1.size(); i++) //通过循环来确定交集

{

for (int j = 0; j < v2.size(); j++)

{

if (v1[i] == v2[j])

{

v3.push\_back(v2[j]); //将交集部分赋值给v3

}

}

}

return v3; //返回vector

}

int main()

{

vector<string>str1, str2;

cout << "请给第一个向量输入五个字符串" << endl;

for (int m = 0; m < 5; m++) //输入五个字符串

{

string m\_str;

cin >> m\_str;

str1.push\_back(m\_str);

}

cout << "请给第二个向量输入五个字符串" << endl;

for (int m = 0; m < 5; m++) //输入五个字符串

{

string n\_str;

cin >> n\_str;

str2.push\_back(n\_str);

}

vector<string>str3(intersection(str1, str2)); //将intersection函数中的v3传递给str3

cout << "两个向量的公共部分为:";

for (vector<string>::iterator it = str3.begin(); it != str3.end(); it++)

{

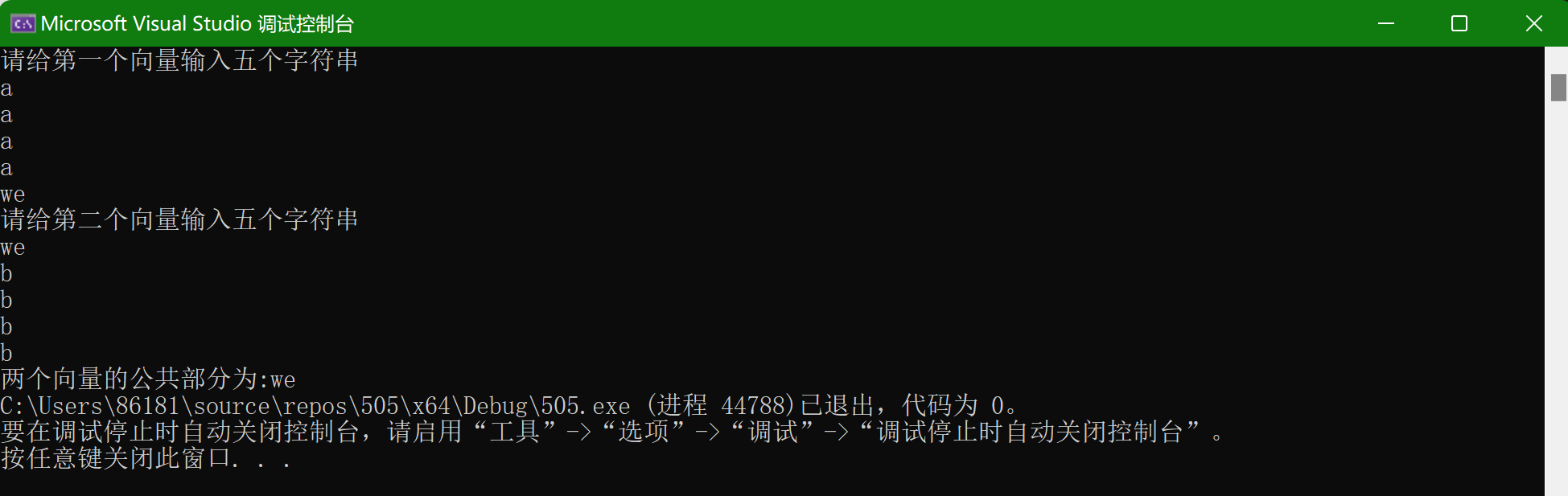
cout << \*it << " ";

}

return 0;

}

结果：



**四、遇到的问题与解决办法**

在写第二个程序时，主函数的cpp开始没有包含student的头文件，导致运行报错，当看到报错提示时才反应过来没有包含student.h。

第四个程序中，所有都写完了，但是求三角形面积的S一直在报错，一开始写的时S\*S== p \* (p - a1) \* (p - b1) \* (p - c1)。但是显示S未初始化后就改变了策略，去csdn搜了根号的表达，然后进行了修改。第四个程序最后的基类指针数组每一个元素指向派生类的对象一开始一直不理解，其实类就是一种自定义的数据类型，该数据类型的具体内容就是该类的对象，那么对象就相当于一个具体的变量数据，指针理应等于指向的数据变量的地址，即指针等于指向的对象的地址。

**五、体会**

类与对象这里重点的内容就是基类与派生类的表示，成员函数类内类外如何定义，虚函数与纯虚函数，虚析构函数与纯虚析构函数，this指针的用法，模板、向量以及容器的用法。我自认这块内容学的一知半解，理解并不深刻，但是语言本来就是趋向于自学自行理解的才能搞明白一门语言，学习路上并不容易，有了老师的加持、csdn的支持，相信最后我一定会打破C++的那层壁。