**计算机程序设计基础（C++）**

**实验报告**

专业班级 软件工程2206班

学号 8209220622

姓名 王昱凯

**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**

1. **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。

1. **实验内容**

1.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?

#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 t1; //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.hour << ":" << t1.minute << ":" << t1.sec << endl;

return 0;

}

2.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(代码已用横线分割)

//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;

};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//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;

}

---------——————————————————————————————

//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(7, "tcg", 'm');

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

return 0;

}

}

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 vector intersection(const vector& v1, const vector& 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。

1. **实验步骤、算法与结果分析**

**第一题：**

**Code：**

#include <iostream>

using namespace std;

class Time {

public:

void setTime(int hour, int minute, int sec) {

this->hour = hour;

this->minute = minute;

this->sec = sec;

}

int getTime() {

cout << "hour: " << this->hour << " minute: " << this->minute << " sec: " << this->sec << endl;

return 0;

}

private:

int hour;

int minute;

int sec;

};

int main()

{

Time t1;

t1.setTime(12, 22, 22);

t1.getTime();

return 0;

}

/\*

1.函数成员应指定为共有的

2.属性成员应指定为私有的

3.将与具体实现无关的函数放在class里面实现；

与具体实现有关的函数放在别的地方实现，在class里只给出声明

\*/

**Result：**

****

**第二题：**

**Code：**

**Student.h**

#pragma once

//#include <string>

class Student {

public:

void set\_Value(int age, const char\* name, const char\* sex);

void display();

private:

int m\_Age;

const char\* m\_Name;

const char\* m\_Sex;

};

**Student.cpp**

#include <iostream>

#include"Student.h"

using namespace std;

void Student::set\_Value(int age, const char\* name,const char\* sex) {

m\_Age = age;

m\_Name = name;

m\_Sex = sex;

}

void Student::display() {

cout << "Age: " << m\_Age<<endl;

cout << "Name:" << m\_Name << endl;

cout << "Sex: " << m\_Sex << endl;

}

**Main.cpp**

#include <iostream>

#include"student.h"

using namespace std;

int main()

{

Student stud;

stud.set\_Value(18, "zhangsan", "m");

stud.display();

return 0;

}

**Result：**

****

**第三题：**

**Code：**

#include <iostream>

using namespace std;

class Cuboid {

public:

void getValue(int length, int width, int height) {

m\_Length = length;

m\_Width = width;

m\_Height = height;

}

int showVolume() {

return m\_Length \* m\_Width \* m\_Height;

}

private:

int m\_Length;

int m\_Width;

int m\_Height;

};

int main()

{

int l1, l2, l3;

int w1, w2, w3;

int h1, h2, h3;

Cuboid c1;

Cuboid c2;

Cuboid c3;

cout << "输入c1的长，宽，高" << endl;

cin >> l1 >> w1 >> h1;

cout << "输入c2的长，宽，高" << endl;

cin >> l2 >> w2 >> h2;

cout << "输入c3的长，宽，高" << endl;

cin >> l3 >> w3 >> h3;

c1.getValue(l1, w1, h1);

c2.getValue(l2, w2, h2);

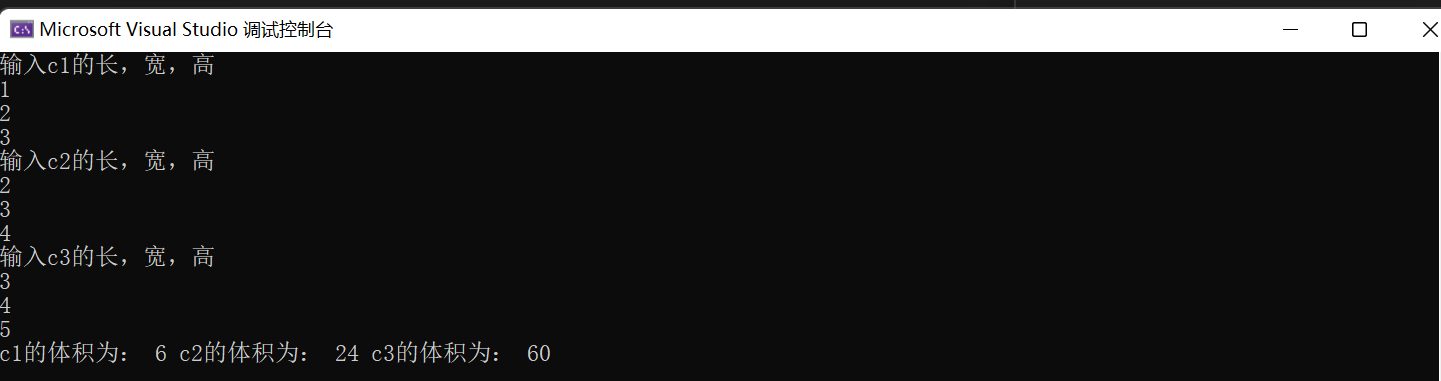
c3.getValue(l3, w3, h3);

cout << "c1的体积为： " << c1.showVolume() << " c2的体积为： " << c2.showVolume() << " c3的体积为： " << c3.showVolume() << endl;

return 0;

}

**Result：**

****

**第四题：**

**Code：**

#include <iostream>

#include"math.h"

#define pi 3.14

using namespace std;

//class Circle;

//class Square;

//class Rectangle;

//class Trapezoid;

//class Triangle;

class Shape {

public:

virtual double getSquare() {

return 0;

}

private:

};

class Circle :public Shape {

public:

Circle(double r) {

m\_R = r;

}

double getSquare() {

return pi \* m\_R \* m\_R ;

}

private:

double m\_R;

};

class Square :public Shape {

public:

Square(double l) {

m\_L = l;

}

double getSquare() {

return m\_L \* m\_L ;

}

private:

double m\_L;

};

class Rectangle :public Shape {

public:

Rectangle(double l, double w) {

m\_Length = l;

m\_Width = w;

}

double getSquare() {

return m\_Length \* m\_Width;

}

private:

double m\_Length;

double m\_Width;

};

class Trapezoid :public Shape {

public:

Trapezoid(double a, double b, double h) {

m\_A = a;

m\_B = b;

m\_H = h;

}

double getSquare() {

return (m\_A + m\_B) \* m\_H \* 0.5 ;

}

private:

double m\_A;

double m\_B;

double m\_H;

};

class Triangle :public Shape {

public:

Triangle(double a, double b, double c) {

m\_A = a;

m\_B = b;

m\_C = c;

}

double getSquare() {

double p;

p = 0.5 \* (m\_A + m\_B + m\_C);

return sqrt(p \* (p - m\_A) \* (p - m\_B) \* (p - m\_C)) ;

}

private:

double m\_A;

double m\_B;

double m\_C;

};

void showSquare(Shape& shape) {

cout<<shape.getSquare()<<endl;

}

int main()

{

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3, 4);

Trapezoid trapezoid(2.0, 4.5, 3);

Triangle triangle(4, 5, 6);

Shape\* p1 = &circle;

Shape\* p2 = &square;

Shape\* p3 = &rectangle;

Shape\* p4 = &trapezoid;

Shape\* p5 = &triangle;

Shape\* a[5] = { p1,p2,p3,p4,p5 };

showSquare(\*a[0]);

showSquare(\*a[1]);

showSquare(\*a[2]);

showSquare(\*a[3]);

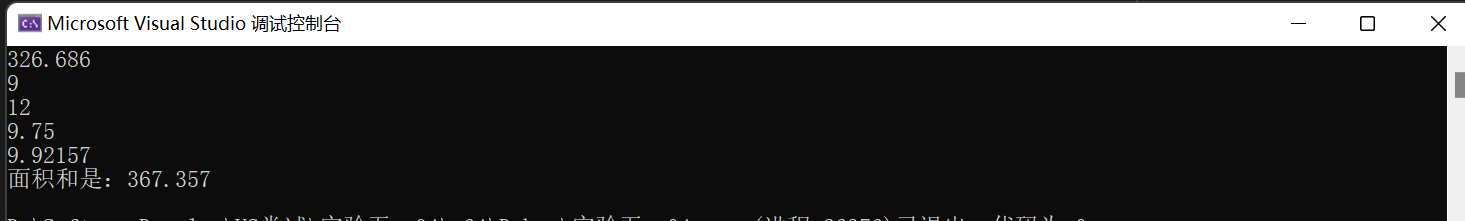
showSquare(\*a[4]);

double sumS = circle.getSquare() + square.getSquare() + rectangle.getSquare() + trapezoid.getSquare() + triangle.getSquare();

cout<<"面积和是："<<sumS<<endl;

}

**Result：**

****

**第五题：**

**Code：**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

template<typename T>

void showVector(const vector<T>& v) {

for (int i = 0; i < v.size(); i++) {

cout << v[i]<<" ";

}

cout << endl;

}

template<typename T>

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

vector<T>v\_result;

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

for (int j = 0; j < 5; j++) {

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

v\_result.push\_back(v1[j]);

}

}

}

return v\_result;

}

int main()

{

cout << "输入v1的五个元素" << endl;

vector<string> v1(5);

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

cin >> v1[i];

}

cout << "-------------------" << endl;

cout << "输入v2的五个元素" << endl;

vector<string> v2(5);

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

cin >> v2[i];

}

vector<string>myVec = intersection(v1, v2);

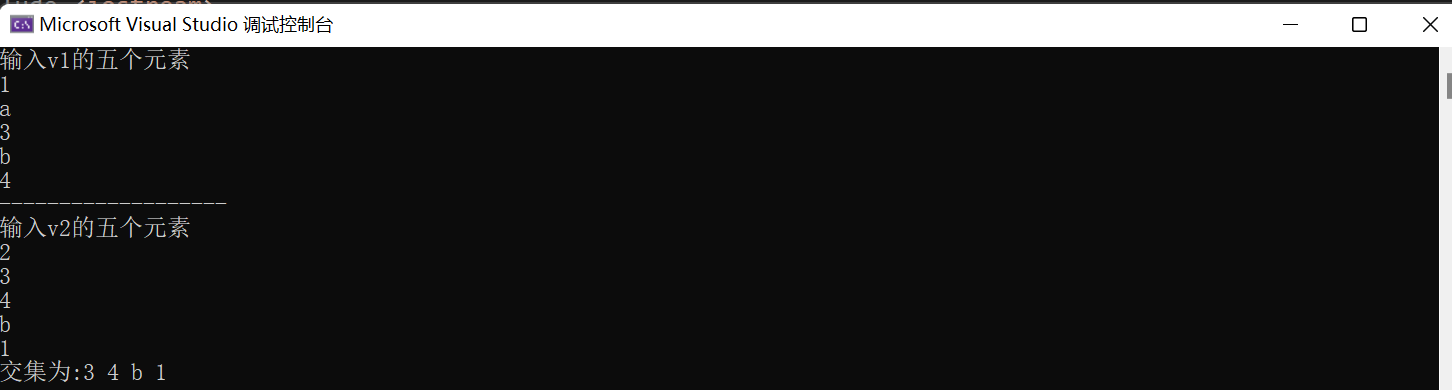
cout << "交集为:";

showVector(myVec);

return 0;

}

**Result：**

****

1. **遇到的问题和解决方法**

**实验5写完template<typename T>vector<T>之后，不知道如何在没有提前知道用户会输入什么数据类型的情况下，能够创建一个vector容器，让vector能够自己判断用户输入的数据类型。我的思路局限于这个容器必须所有的元素属于同一个数据类型，但是又无法实现让vector自己判断。解决方法是在csdn上查找，突然想到可以把vector实例化为string类型，从而使伪命题得以解决**

1. **体会**

**类与对象是c++面向对象编程的重要特征，面向对象的编程的特性就是封装性，继承性，多态性。通过实验五，我对面向对象的编程有了更深的理解，并深深的体会到了面向对象编程的优势优点。同时也让我对STL，多态，虚函数等不太熟悉的知识点有了更深的理解**