**实验五、类与对象**

**一、实验目的与要求**

**【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。

**二、实验内容**

**【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;14

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、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。

1. **算法分析、程序与结果**

**1程序**

#include<iostream>

using namespace std;

class Time // 定义 Time 类

{

public:

void setTime()

{

cin >> hour;

cin >> minute;

cin >> sec;

}

void showTime()

{

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

}

private:// 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

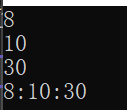
t1.setTime();

t1.showTime();

return 0;

}

**结果**



2程序

//student.h

#include<iostream>

using namespace std;

class Student

{

public:

void display();

private:

string num;

string name;

char sex;

public:

void set\_value() {

string a;

cout << "输入学号：";

cin >> a;

num = a;

cout << "输入姓名：";

cin >> name;

char c;

cout << "输入性别：";

cin >> c;

sex = c;

}

Student() {

this->num = num;

this->name = name;

this->sex = sex;

}

Student(string a, string b, char c) {

num = a;

name = b;

sex = c;

}

};

//student.cpp

#include<iostream>

using std::cout; using std::endl;

#include"student.h"

void Student::display()

{

cout << "num:" << num << endl;

cout << "name:" << name << endl;

cout << "sex:" << sex << endl;

}

//main.cpp

#include<iostream>

#include<string>

#include"student.h"

int main() {

Student stud;

stud.set\_value();

stud.display();

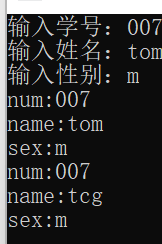
Student stud1("007", "tcg", 'm');

stud1.display();

return 0;

}

结果



3程序

#include<iostream>

using namespace std;

class Rectangular {

private:

int length;

int width;

int height;

public:

void setRectangular(){

int a, b, c;

cout << "输入长方体的长=";

cin >> a;

cout << "输入长方体的宽=";

cin >> b;

cout << "输入长方体的高=";

cin >> c;

this->length = a;

this->width = b;

this->height = c;

}

void getRectangular() {

int V = length \* width \* height;

cout<<"长方体体积="<<V<<endl;

}

};

int main() {

Rectangular r1;

r1.setRectangular();

r1.getRectangular();

Rectangular r2;

r2.setRectangular();

r2.getRectangular();

Rectangular r3;

r3.setRectangular();

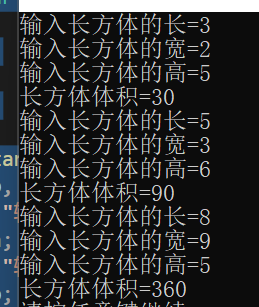
r3.getRectangular();

system("pause");

return 0;

}

结果



4程序

#include<iostream>

#include<math.h>

using namespace std;

class Shape {

public:

virtual double area() = 0;

virtual void display() = 0;

};

class Circle :public Shape {

private:

double m\_r;

public:

Circle(double r) {

this->m\_r = r;

}

virtual double area() {

return 3.14 \* m\_r \* m\_r;

}

void display() {

cout << "圆的面积=" << this->area()<<endl;

}

};

class Square :public Shape {

private:

double m\_a;

public:

Square(double a) {

this->m\_a = a;

}

virtual double area() {

return m\_a \* m\_a;

}

void display() {

cout << "正方形的面积=" << this->area()<<endl;

}

};

class Rectangle:public Shape {

private:

double m\_a;

double m\_b;

public:

Rectangle(double a, double b) {

this->m\_a = a;

this->m\_b = b;

}

virtual double area() {

return m\_a \* m\_b;

}

void display() {

cout << "长方形的面积=" << this->area() << endl;

}

};

class Trapezoid :public Shape {

private:

double m\_a;

double m\_b;

double m\_h;

public:

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

this->m\_a = a;

this->m\_b = b;

this->m\_h = h;

}

virtual double area() {

return (m\_a + m\_b)\*m\_h/2;

}

void display() {

cout << "梯形的面积=" << this->area() << endl;

}

};

class Triangle :public Shape {

private:

double m\_a;

double m\_b;

double m\_c;

public:

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

this->m\_a = a;

this->m\_b = b;

this->m\_c = c;

}

virtual double area() {

double p = (m\_a + m\_b + m\_c) / 2;

return sqrt(p\*(p-m\_a)\*(p-m\_b)\*(p-m\_c)) ;

}

void display() {

cout << "三角形的面积=" << this->area() << endl;

}

};

int main() {

Circle circle(10.3);

circle.area();

circle.display();

Square square(3);

square.area();

square.display();

Rectangle rectangle(3,4);

rectangle.area();

rectangle.display();

Trapezoid trapezoid(2.0,4.5,3);

trapezoid.area();

trapezoid.display();

Triangle triangle(4,5,6);

triangle.area();

triangle.display();

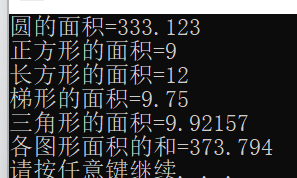
cout << "各图形面积的和="<<circle.area() + square.area()+ rectangle.area() + trapezoid.area() + triangle.area() << endl;

system("pause");

return 0;

}

结果



5程序

#include<iostream>

using namespace std;

class Time // 定义 Time 类

{

public:

void setTime()

{

cin >> hour;

cin >> minute;

cin >> sec;

}

void showTime()

{

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

}

private:// 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

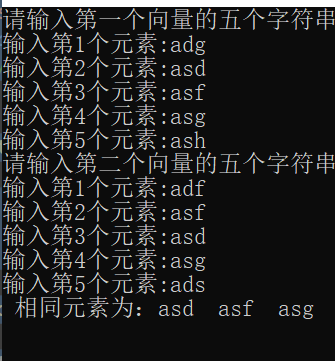
t1.setTime();

t1.showTime();

return 0;

}

结果



1. 遇到的问题与解决办法

在编译的时候总是会遇到一些报错例如无法解析main， "expression" 后出现意外标记 "标识符" 等问题，通过我对错误警告的理解和上网查找解决办法最后解绝

五、体会

这次实验不仅有类与对象方面的问题，还有一些STL和模板方面的知识点，将这些知识点融会贯通才能更好的进行c++ 的编程，对于这些问题我还要进行更多的练习