**实验五、类与对象**

**一、【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】**

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、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、算法与结果分析**

1、程序：

#include <iostream>

using namespace std;

class Time

{

private:

int hour;

int minute;

int sec;

public:

Time(int hour=0,int minute=0,int sec=0)

{

this->hour=hour;

this->minute=minute;

this->sec=sec;

}

void Showtime()

{

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

}

};

int main()

{

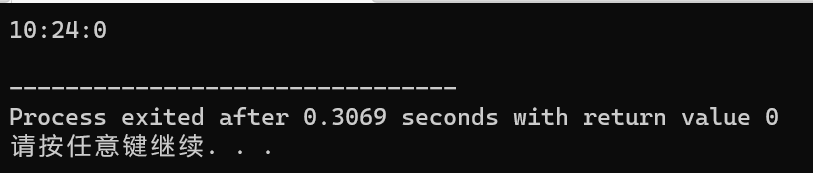
Time t1(10,24,00);

t1.Showtime();

return 0;

}

结果：



2、程序：

（1）头文件：

#ifndef SOMETHING\_H

#define SOMETHING\_H

class Student {

private:

int num;

char name[20];

char sex;

// Private section

public:

void display();

Student(int num=000,char name[20]="NULL",char sex='m')

{

this->num=num;

for(int i=0;i<20;i++){

this->name[i]=name[i];

}//写成this->name[20]=name[20]就只给第二十一个数组元素赋值了

this->sex=sex;

}// Public Declarations

protected:

// Protected Declarations

};

#endif

（2）源文件：

#include <iostream>

#include "student.h"

using namespace std;

void Student::display()

{

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

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

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

}

int main()

{

Student stud;//未初始化，默认使用构造函数

Student stud1(007,"tcg",'m');//给定数据初始化

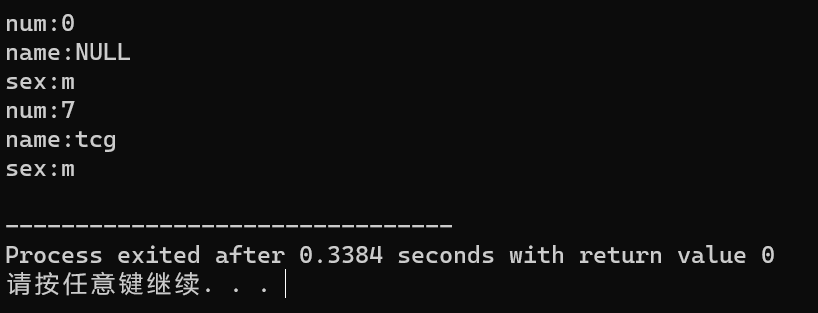
stud.display();

stud1.display();

return 0;

}

结果：



3、程序：

#include <iostream>

using namespace std;

class cuboid

{

private:

double height;

double length;

double width;

public:

cuboid(double height=0,double length=0,double width=0)

{

this->height=height;

this->length=length;

this->width=width;

}

void Setdata(){

cin >> height;

cin >> length;

cin >> width;

}

double calculateV(){

double V;

V=height\*length\*width;

cout << "V=" << V << endl;

return 0;

}

};

int main()

{

cuboid A1;

A1.Setdata();

cuboid A2;

A2.Setdata();

cuboid A3;

A3.Setdata();

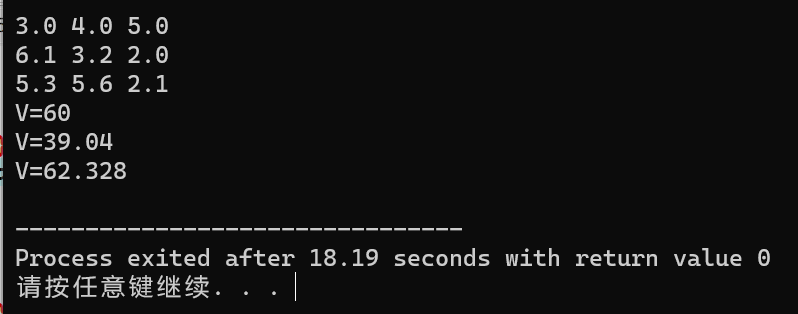
A1.calculateV();

A2.calculateV();

A3.calculateV();

return 0;

}

结果：  


4、程序：

#include <iostream>

#include <cmath>

const double PI=3.14;

using namespace std;

class Shape

{

public:

virtual double area()=0;

virtual ~Shape(){

};

};

class Circle:public Shape

{

private:

double r;

public:

Circle(double r)

{

this->r=r;

}

double area()

{

return PI\*r\*r;

}

~Circle(){

};

};

class Square:public Shape

{

private:

int d;

public:

Square(int d)

{

this->d=d;

}

double area()

{

return d\*d;

}

~Square(){

};

};

class Rectangle:public Shape

{

private:

int a;

int b;

public:

Rectangle(int a,int b)

{

this->a=a;

this->b=b;

}

double area()

{

return a\*b;

}

~Rectangle(){

};

};

class Trapezoid:public Shape

{

private:

double a,b,c;

public:

Trapezoid(double a,double b,double c)

{

this->a=a;

this->b=b;

this->c=c;

}

double area()

{

return (a+b)\*c/2;

}

~Trapezoid(){

};

};

class Triangle:public Shape

{

private:

int a,b,c;

public:

Triangle(int a,int b,int c)

{

this->a=a;

this->b=b;

this->c=c;

}

double area()//进行三角形面积计算时判定三角形是否成立

{

if(a+b>c&&a+c>b&&b+c>a)

{

double S;

S=sqrt((a+b+c)\*(a+b-c)\*(a+c-b)\*(b+c-a))/4;

return S;

}

else

{

cout << "error" << endl;

return 0;

}

}

~Triangle(){

};

};

int main()

{

Shape \* A[5];

A[0]=new Circle (10.2);//circle类对象进行初始化;

A[1]=new Square (3);//square类对象进行初始化

A[2]=new Rectangle (3,4);//rectangle类对象进行初始化

A[3]=new Trapezoid (2.0,4.5,3);//trapezoid类对象进行初始化

A[4]=new Triangle (4,5,6);//triangle类对象进行初始化

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

{

cout << A[i]->area() << endl;

//delete A[i];如果在这里就释放了内存，后面面积求和无法返回结果

}

double sum=0;

double s[5];

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

s[i]=A[i]->area();

sum=sum+s[i];

delete A[i];

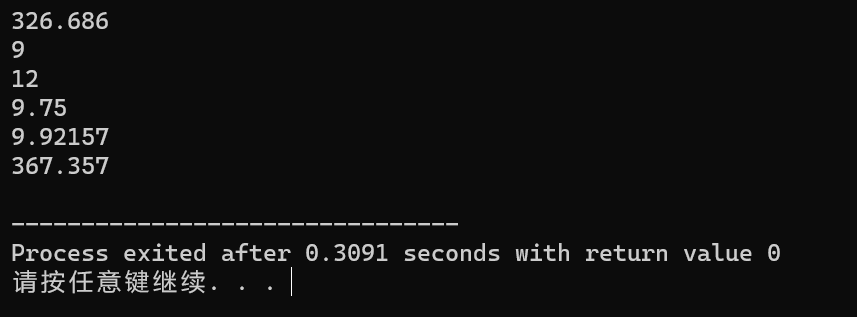
}

cout << sum << endl;

return 0;

}

结果：



5、程序：

#include <iostream>

#include <string>

#include <vector>

using namespace std;

template <class T>

//找两个vector容器中的交集，每个容器中有五个字符串

vector <T> intersection(vector<T>& v1,vector<T>& v2,vector<T> v3){

class vector<T>::iterator it;

class vector<T>::iterator it1;

for(it=v1.begin();it!=v1.end();it++){

for(it1=v2.begin();it1!=v2.end();it1++){

if(\*it==\*it1){

v3.push\_back(\*it1);//如果v2中元素与v1一致就放入v3容器中

};

};

};

for(int j=0;j<v3.size();j++){

cout << v3[j] << "\t";

}

return v3;

}

int main()

{

vector <string> v1,v2,v3;

string str1[5];

string str2[5];

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

cin >> str1[i];

}

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

cin >> str2[j];

}//注意string的输入方式：如果写成for(int i=0;i<5;i++){

// cin >> str1[i];

// cin >> str2[i];}则是交替将输入的数据分别存入两个数组中；

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

v1.push\_back(str1[j]);

v2.push\_back(str2[j]);

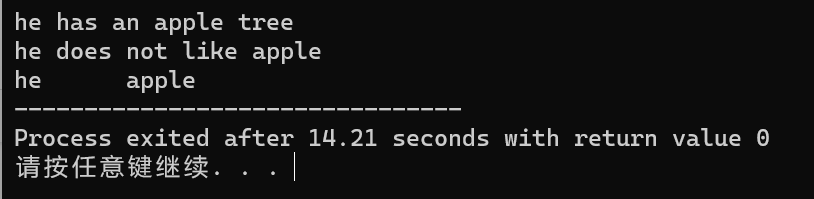
}

intersection(v1,v2,v3);

return 0;

}

结果：



**四、遇到的问题与解决方法**

这次实验遇到的问题集中在第五道实验中，首先是因为对STL的不熟悉，定义模板时使用typename <T>后程序报错 …is not a class or struct。之后将typename改成class后问题解决。然后又遇到了一个我用了很久才解决但实际上是个低级问题的问题，在输入string数组时因为在同一个for循环内同时输入，然后以为是先输入str1再输入str2，但是实际上是交替输入，导致最后的结果始终不对，最终在数个小时的纠错中终于发现了错误，最终程序可以正常运行。

**五、心得与体会**

刚学的知识就拿去做实验非常不熟练，还是要课后多看看书然后自己动手实践，加深理解，做到可以灵活使用，有些细节就要靠自己的积累和细心了，不应该想当然，可以把类似的问题简化后动手实践，验证一下是否正确，可以提高代码的正确率。