**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 8209220621

姓 名 陈墨

**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**

**【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、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。

**【算法分析、程序与结果】**

1. 程序：

#include<iostream>

using namespace std;

class Time

{

private:

int hour;

int minute;

int second;

public:

void tinsert()

{

int h,m,s;

cout << "请依次输入小时，分钟，秒" << endl;

cin >> h >> m >> s;

hour = h;

minute = m;

second = s;

}

void tshow()

{

cout << "时间是" << endl;

cout << hour << ":" << minute << ":" << second;

}

};

int main()

{

Time t1;

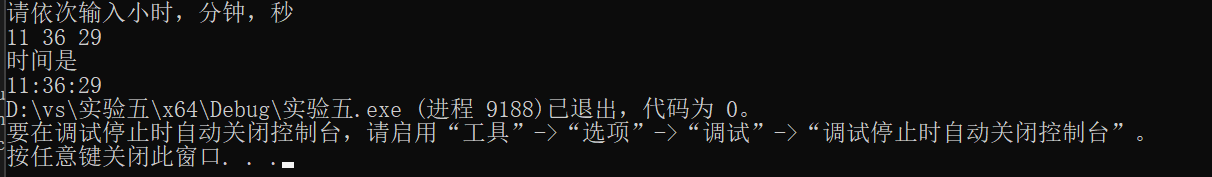
t1.tinsert();

t1.tshow();

return 0;

}

运行结果:



2、程序：

student.h:

class Student {

private:

int num;

char name[20];

char sex;

public:

void display();

void set\_value(int n,const char nm[],const char s);

};

student.cpp:

#include<iostream>

#include<string>

#include"student.h"

using namespace std;

void Student::display()

{

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

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

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

}

void Student::set\_value(int n, const char nm[],const char s)

{

num = n;

strcpy\_s(name, nm);

sex = s;

}

5.2.cpp:

#include<iostream>

#include"student.h"

int main()

{

Student stud;

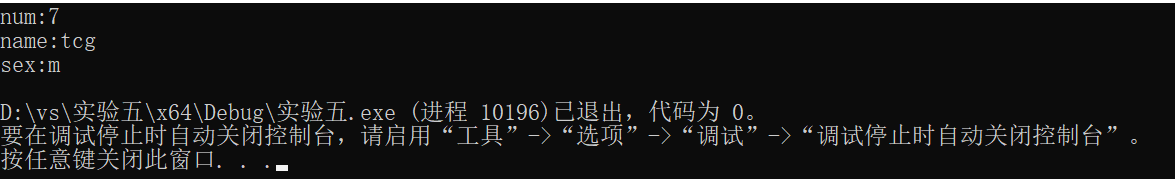
stud.set\_value(007, "tcg", 'm');

stud.display();

return 0;

}

运行结果:



3、程序：

#include<iostream>

using namespace std;

class Cuboid {

private:

double length;

double width;

double height;

public:

void insert()

{

int l, w, h;

cin >> l >> w >> h;

length = l;

width = w;

height = h;

}

double c\_volume()

{

return height \* length \* width;

}

void c\_v\_display()

{

cout << c\_volume()<<endl;

}

};

int main()

{

Cuboid c1, c2, c3;

c1.insert();

c2.insert();

c3.insert();

c1.c\_v\_display();

c2.c\_v\_display();

c3.c\_v\_display();

return 0;

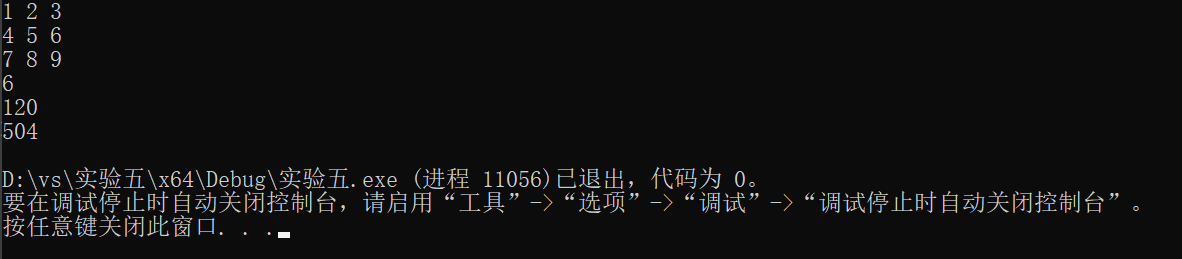
}

分别输入1 2 3

4 5 6

7 8 9

结果如下：



4、程序：

#include<iostream>

#include<cmath>

using namespace std;

class Shape {

public:

virtual double area() = 0;

};

class Circle :public Shape {

private:

double radius;

public:

Circle(double r)

{

radius = r;

}

double area()

{

return 3.14 \* pow(radius, 2);

}

};

class Square :public Shape{

private:

double side;

public:

Square(double s)

{

side = s;

}

double area()

{

return pow(side, 2);

}

};

class Rectangle :public Shape{

private:

double width;

double length;

public:

Rectangle(double w, double l)

{

width = w;

length = l;

}

double area()

{

return width \* length;

}

};

class Trapezoid :public Shape{

private:

double top;

double bottom;

double height;

public:

Trapezoid(double t, double b, double h)

{

top = t;

bottom = b;

height = h;

}

double area()

{

return 1 / 2 \* (top + bottom) \* height;

}

};

class Triangle :public Shape{

private:

double side1;

double side2;

double side3;

public:

Triangle(double a, double b, double c)

{

side1 = a;

side2 = b;

side3 = c;

}

double area()

{

double t = (side1 + side2 + side3)/2;

return sqrt(t\*(t - side1)\*(t - side2)\*(t - side3));

}

};

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

double area\_sum=0;

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

{

area\_sum += shapearr[i]->area();

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

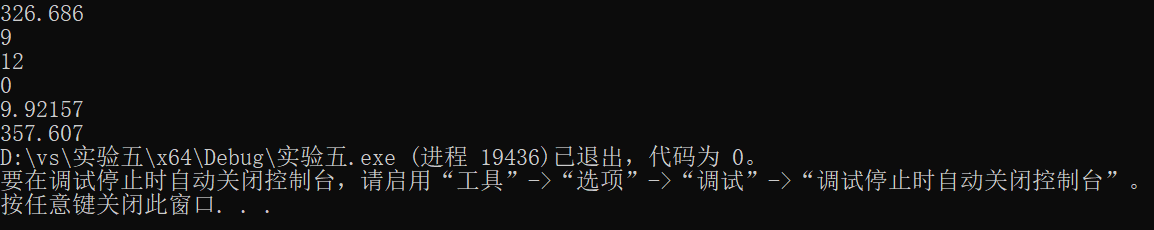
}

cout << area\_sum;

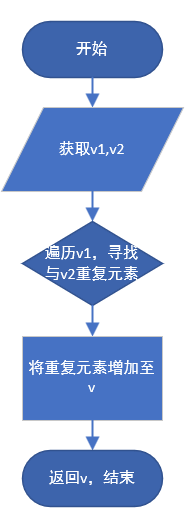
return 0;

}

运行结果：



5、算法：



程序：

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

template<typename T>

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

{

vector<T> v;

for (int i = 0; i < v1.size(); i++)

{

if (find(v2.begin(), v2.end(), v1[i]) != v2.end())

{

v.push\_back(v1[i]);

}

}

return v;

}

int main() {

vector<string> v1(5), v2(5);

cout << "Enter 5 strings for vector 1:" << endl;

for (int i = 0; i < v1.size(); i++)

{

cin >> v1[i];

}

cout << "Enter 5 strings for vector 2:" << endl;

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

{

cin >> v2[t];

}

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

cout << "The intersection of the two vectors is:" << endl;

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

{

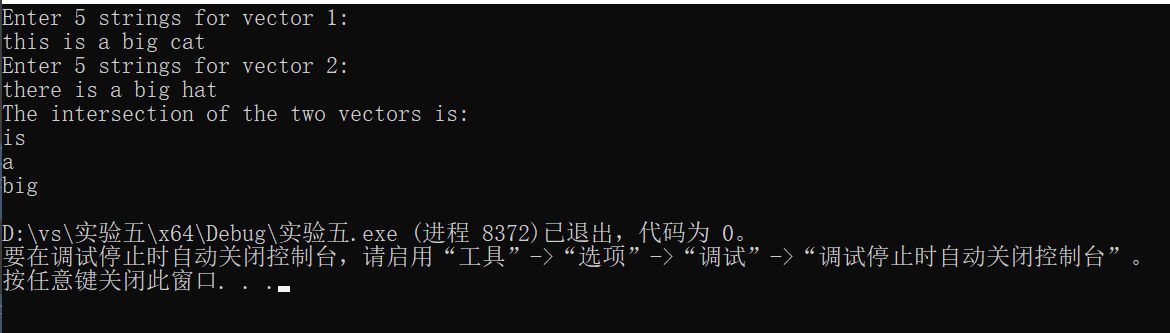
cout << intersect[i] << endl;

}

return 0;

}

运行结果：



**【遇到的问题及解决办法】**

前四个程序整体上来说，都比较常规，没有太多思考与理解的难度，无非是比较繁琐，细致一点就问题不大，最后一个程序的话，由于网课学习的效率不佳，刚上手时候不能很好理解，在各个网站再进行了一遍相对细致的学习，对各个STL及函数模板有了更好的理解，解决这个问题便也简单起来，使用一个for循环来遍历对比即可，最后也是成功得到结果。

**【体会】**

这次实验下来，对类以及STL都有了更好的理解与掌握，虽然遇到了很多小问题，但都成功解决了，整体来说，还是收获满满，这虽然是最后一次实验，但C++的学习还有很长一段路要走，期待以后能对C++进一步学习了解。