**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018-09-18 指导教师签字： 刘永芬 成绩：

**实验1 C++基础练习（一）**

一、实验目的和要求

（1）掌握C++程序的基本格式与规范，学会编写简单的C++程序，理解C++程序结构的特点。

（2）熟悉C++的非面向对象语言特征及其使用方法。

二、实验内容和原理

1、输入并运行下面的程序，改正所出现的错误信息，并写出输出结果。

int main();

{int a,b;

c=a+b;

cout>>"a+b=">>a+b;}

如果出错，则根据编译信息指出的错误，修改程序，再进行编译，如果还有错，再重复此过程，直到编译不出错为止。

2、首先编写函数long convert(int d,int h,int m)，函数的功能是实现将d天h小时m分钟换算成分钟，然后在主函数main（）中实现输入和输出。

3、打印出所有的“水仙花”数，所谓“水仙花”是指一个3位数，其各位数字立方和等于该数本身。例如153是个水仙花数，因为153=13+53+33。

三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1.打开VC++6.0，新建一个文件，然后把报告里的代码复制粘贴过去，接着运行这个程序。程序出错，根据编译信息指出的错误，进行修改程序，接着再编译程序，发现错误再修改，直至程序正常运行。代码：#include<iostream>

using namespace std;

int main()

{int a,b,c;

cin>>a>>b;

c=a+b;

cout<<"a+b="<<a+b<<endl;

return 0;

}

2.long convert 函数主要实现将d天h小时m分钟换算成分钟。在主程序中定义实参天，时，分，然后调用long convert将实参传给形参，long convert函数返回天\*24\*60+时\*60+分，最后在主函数中输出。代码：#include<iostream>

using namespace std;

long convert(int d,int h,int m)

{

int c;

c=d\*24\*60+h\*60+m;

return c;

}

int main(){

int x,y,z;

scanf("%d%d%d",&x,&y,&z);

cout<<convert( x, y, z)<<endl;

return 0;

}

3.定义三个int型整数，分别用于表示个十，百，位数，用一个for循环找出100到999之间的水仙花数，在for循环里，个位数= i%100%10，十位数=i%100/10，百位数= i/100。接着用if语句判断个位数的三方+十位数的三方+百位数的三方是否等于三位数本身，如果=，输出这个三位数。代码：#include<iostream>

#include<cmath>

using namespace std;

int main(){

int i,a=0,b=0,c=0;

for(i=1;i<1000;i++){

a=i%100%10;

b=i%100/10;

c=i/100;

if(pow(a,3)+pow(b,3)+pow(c,3)==i)

{

cout<<i<<"是水仙花数"<<endl;

}

}

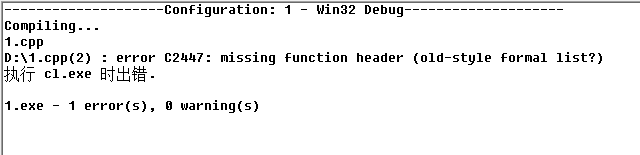
return 0;

}

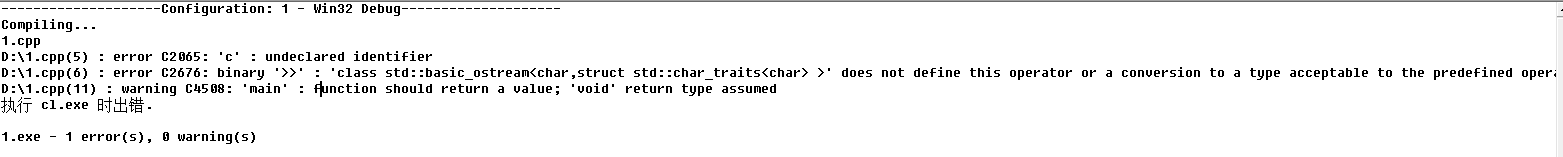
五、调试过程

1.

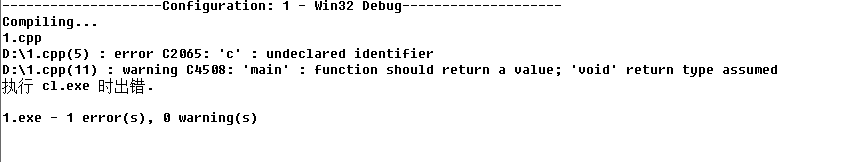
第一次调试报错，缺少头文件



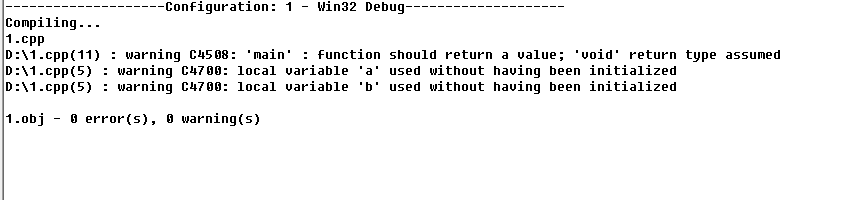
第二次调试时报错，cout标识符出错



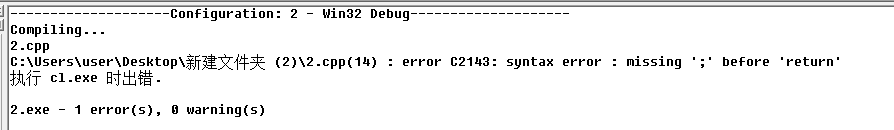
第三次调试时报错，变量c未申明



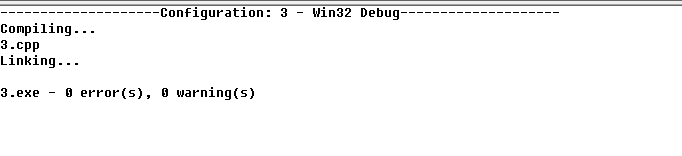
第四次调试时报错，标量a,b未申明，程序无返回值



2.调试错误，少了分号

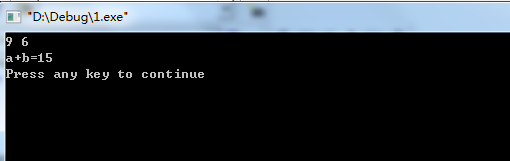


3.调试结果正确



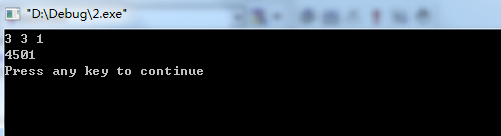
六、实验结果

1.



输入a和b分别是9和6，结果是15正确

2.



输入天，时，分，分别是3，3，1，输出4501分，结果正确

3.



程序输出100到999之间的水仙花数，结果正确。

七、总结

通过这一次的c++基础练习的上机实验，我学习到了许多的知识，也感觉到自己还有很多的不足，特别是基础的知识薄弱，意识到自己还要学习的东西有很多，今后一定更加努力学习新知识，并不断地复习巩固以前的知识。

**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018-09-25 指导教师签字： 刘永芬 成绩：

**实验1 C++基础练习（二）**

一、实验目的和要求

（1）掌握C++程序的基本格式与规范，学会编写简单的C++程序，理解C++程序结构的特点。

（2）熟悉C++的非面向对象语言特征及其使用方法。

二、实验内容和原理

1、定义函数实现计算1～ｎ之和，调用函数输出结果。

2、编写重载函数min可分别求取2个整数、3个整数、2个双精度数和3个双精度数的最小值。

3、随机生成10个（0～100）的整数，并按从小到大的顺序进行排序，输出结果。

三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1. #include<iostream>

using namespace std;

int sum(int n){

int i,m=0;

for(i=1;i<=n;i++)

m=m+i;

return m;

}

int main(){

int n,i;

cout<<"请输入n"<<endl;

cin>>n;

cout<<sum(n)<<endl;

return 0;

}

2. #include<iostream>

using namespace std;

int min(int x,int y){

int t;

if(x<y)

t=x;

else t=y;

return t;

}

int min(int x,int y,int z){

int t;

if(x<y) t=x; if(t>z) t=z;

return t;

}

double min(double x,double y){

double t;

if(x<y)

t=x;

else t=y;

return t;

}

double min(double x,double y,double z){

double t;

if(x<y) t=x; if(t>z) t=z;

return t;

}

int main(){

while (1){

int a,b,c,d,e;

double h,i,j,k,l;

cout<<"按1求取两个整数的最小值\n按2求三个整数的最小值\n按3求两个双精度整数的最小值\n按4求三个双精度整数的最小值\n按9结束程序"<<endl;

int o;

cin>>o;

if(o==1){

cout<<"请输入两个整数：";

cin>>a>>b;

cout<<min( a, b)<<endl;}

if(o==2){

cout<<"请输入三个整数：";

cin>>c>>d>>e;

cout<<min(c,d,e)<<endl;}

if(o==3){

cout<<"请输入两个双精度整数：";

cin>>h>>i;

cout<<min(h,i)<<endl;}

if(o==4){

cout<<"请输入三个双精度整数：";

cin>>j>>k>>l;

cout<<min(j,k,l)<<endl;}

if(o==9)

break;

return 0;

}

3. #include<iostream>

#include<stdlib.h>

#include<time.h>

using namespace std;

int main(){

int i,j,t;

int a[10];

srand((int)time(0));

for(i=0;i<10;i++){

a[i]=(int )(0+100\*rand()/RAND\_MAX);

printf("%5d",a[i]);

}

printf("\n");

for(i=0;i<9;i++)

for(j=0;j<9;j++)

if(a[j]>a[j+1]){

t=a[j];

a[j]=a[j+1];

a[j+1]=t;

}

printf("数列排序为：\n");

for(i=0;i<10;i++)

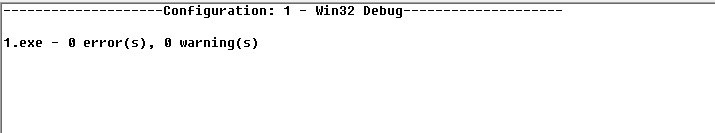
printf("%5d",a[i]);

return 0;

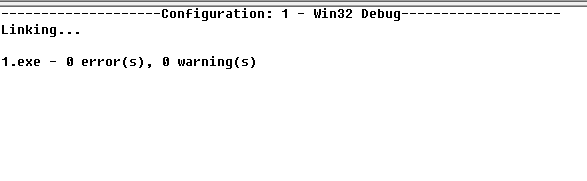
}

五、调试过程

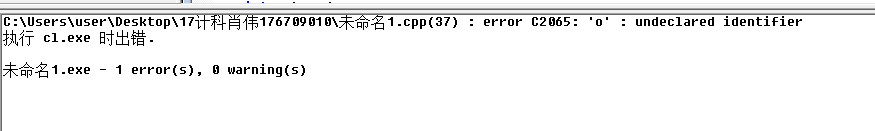
1.第一次调试，编译通过，但1到n的和m未初始化为0



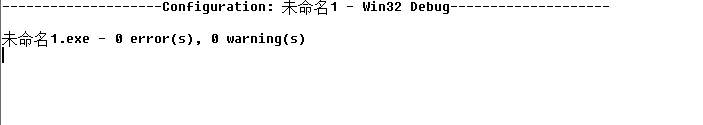
第二次调试，编译通过，程序正常运行



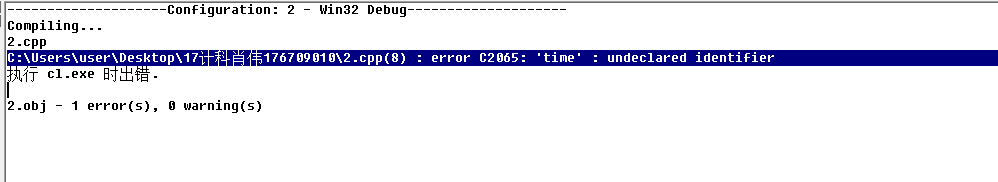
2.第一次编译选择变量o未申明



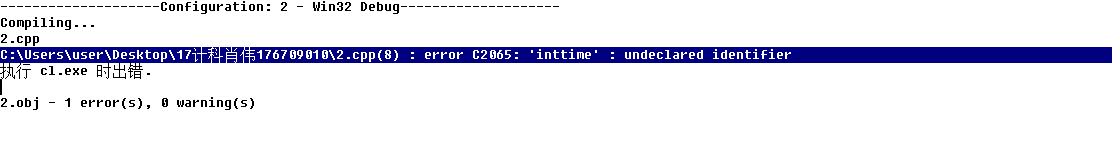
第二次编译通过



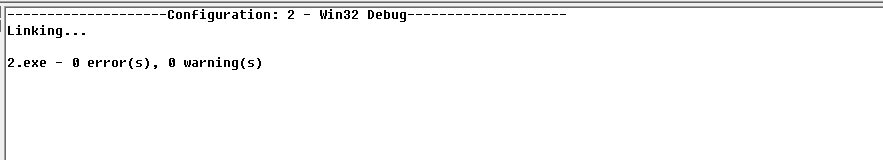
3.第一次调试，编译错误，原因：产生随机数的函数头文件未包含在程序中



第二次调试，编译错误，原因：srand()函数产生的伪随机数类型未定义

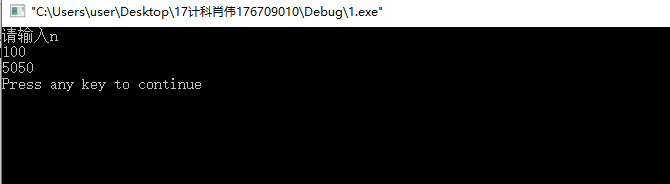


第三次调试，编译通过，程序正常运行



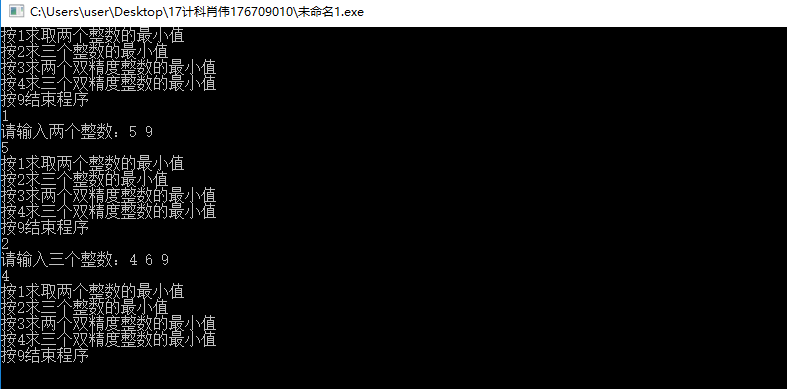
六、实验结果

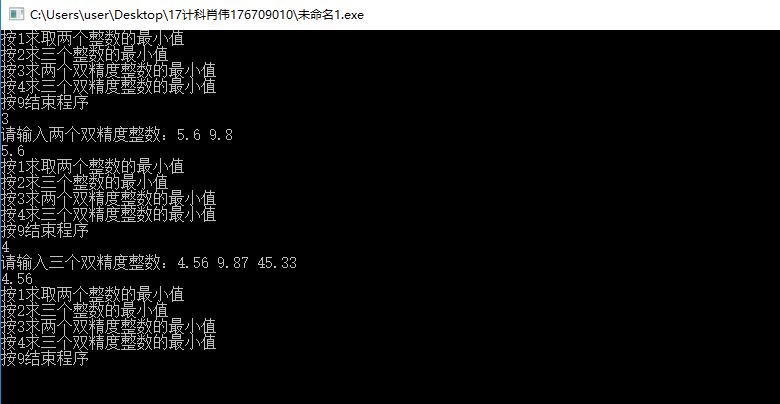
1.



输入n=100，输出1加到100的结果5050程序正确。

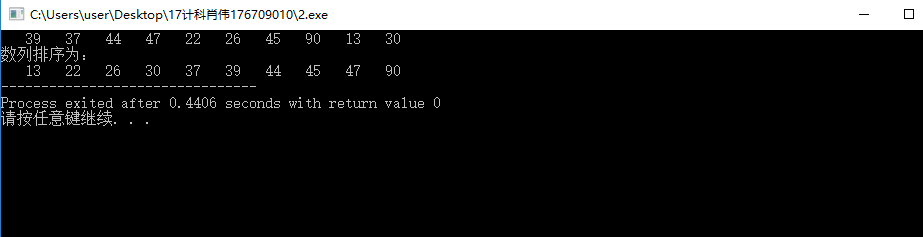
2.





按提示输入数字进行相关操作，按1输入两个整型数字，5 9，输出最小值5正确，按2输入三个整型数字4 6 9，输出最小值4正确，以此类推，程序正确。

3.



产生1个随机数并输出屏幕，然后在按从小到大的顺序输出，如图所示，结果正确。

七、总结

通过这次的c++基础上机实验，巩固了以往的c语言基础知识，加深了印象，并深刻意识到自己以往学习中的不足，比如对之前的编写格式算法的的遗忘导致了编程频繁出错，今后我会更加努力学习新知识，并在课余时间补以前的不足之处。

**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018.9.9 指导教师签字： 刘永芬 成绩：

**实验2 类和对象（一）**

一、实验目的和要求

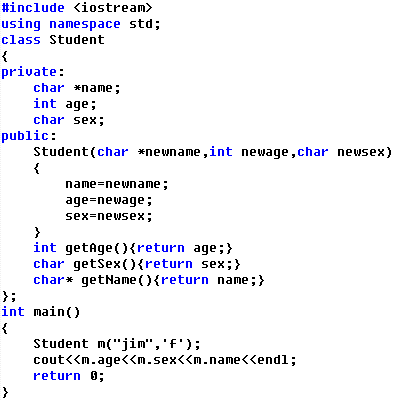
(1) 掌握类、类的数据成员、类的成员函数的定义方式，理解类成员的访问控制方式。

(2) 掌握对象的定义和操作对象的方法。

(3) 掌握构造函数和析构函数的定义与执行过程。

二、实验内容和原理

1、调试下面程序，在不删除和增加语句的情况下，改正错误语句，使其正确运行。



2、(1)在以下程序中，将TPoint类的带有两个参数的构造函数进行修改，在函数体内增加如下语句：

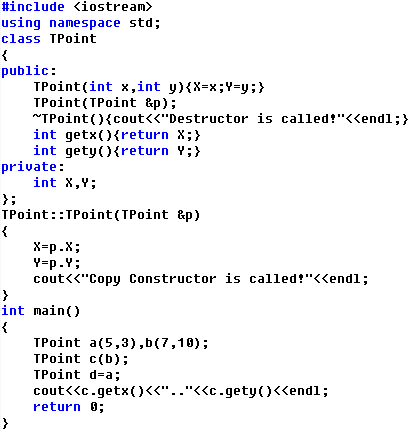
cout<<”Constructor is Called! ”<<endl;

(2)写出程序的输出结果，并解释输出结果。

(3)在主函数体内，添加下列说明语句：

TPoint e(20),f;

调试程序会出现什么现象？为什么？如何解决？



3、请定义一个三角形类triangle，私有数据成员为三角形的三边（first,second,third），设计无参构造置first、second、third都为0，有参构造函数置三边为对应形参的值，另外还包括求三角形周长、求三角形的面积、修改三角形三边的值，输出三角形三边等公有成员函数。要求输出三角形三边的格式为“三边分别是：first、second、third”。

三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1. #include<iostream>

using namespace std;

class Student

{

private:

char \*name;

int age;

char sex;

public:

Student(char \*newname,int newage,char newsex)

{

name=newname;

age=newage;

sex=newsex;

}

int getAge(){return age;

}

char getSex(){return sex;

}

char\* getName(){return name;

}

};

int main(){

Student m("jim",18,'f');

cout<<m.getAge()<<m.getSex()<<m.getSex()<<endl;

}

2.

#include <iostream>

using namespace std;

class TPoint

{

public:

TPoint(int x,int y){

X=x;Y=y;

cout<<"Constructor is Called! "<<endl;

}

TPoint(TPoint &p);

~TPoint(){cout<<"Destructor is called!"<<endl;}

int getx(){return X;}

int gety(){return Y;}

private:

int X,Y;

};

TPoint::TPoint(TPoint &p)

{

X=p.X;

Y=p.Y;

cout<<"Copy Constructor is called!"<<endl;

}

int main()

{

TPoint a(5,3),b(7,10);

TPoint c(b);

TPoint d=a;

cout<<c.getx()<<".."<<c.gety()<<endl;

return 0;

}

3.

#include<iostream>

#include<cmath>

using namespace std;

class Triangle

{

public:

Triangle(int a, int b, int c);

void display();

Triangle();

private:

int first,second,third;

};

Triangle::Triangle()

{

first=0;

second=0;

third=0;

display();

}

Triangle::Triangle(int a, int b, int c)

{

first=a;

second=b;

third=c;

int s;

double area, p;

p=double(0.5\*(a+b+c));

area=sqrt(p\*(p-a)\*(p-b)\*(p-c));

s=first+second+third;

display();

cout<<"周长是"<<s<<endl;

cout<<"面积是"<<area<<endl;

}

void Triangle::display()

{

cout<<"三边分别是:"<<first<<" "<<second<<" "<<third<<" "<<endl;

}

int main()

{

Triangle p1;

Triangle p2(3,4,5);

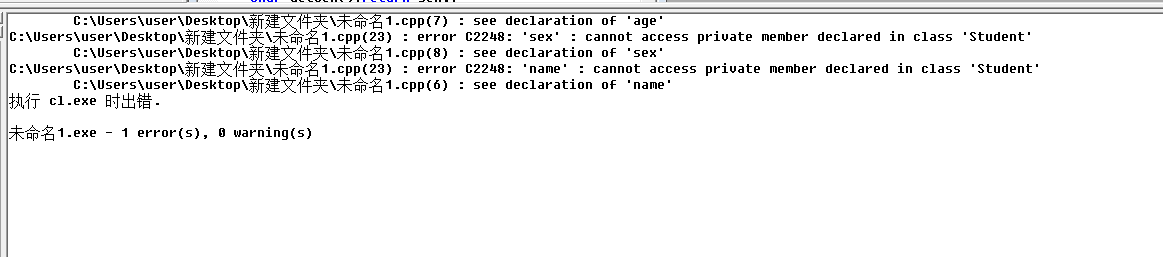
return 0;

}

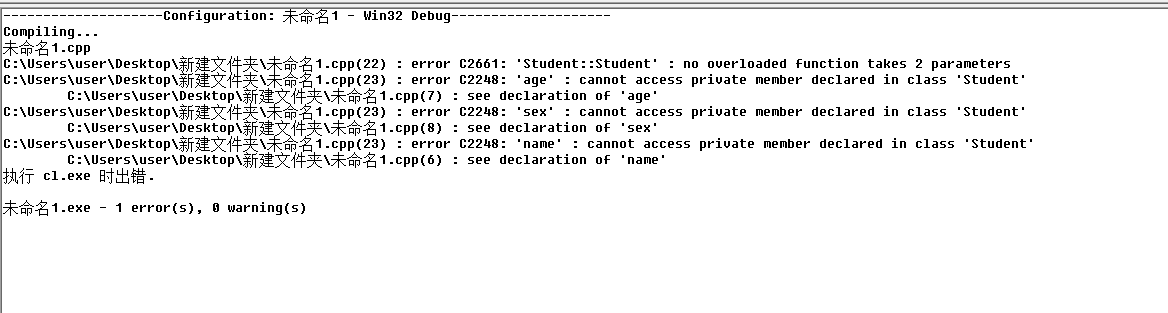
五、调试过程

1．

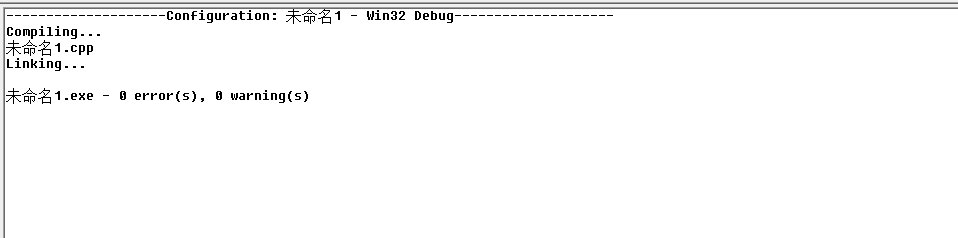
第一次调试：缺少输入年龄



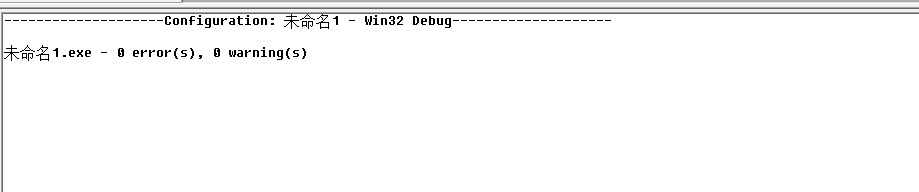
第二次调试：无法访问类成员数据



第三次调试：程序正常，无报错

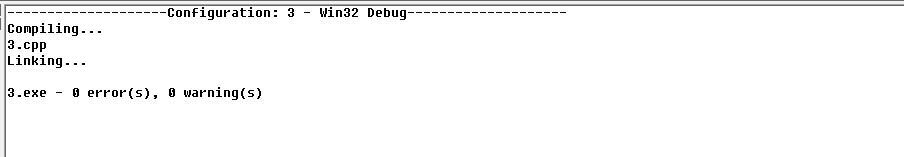


2.调试：程序正常，无报错

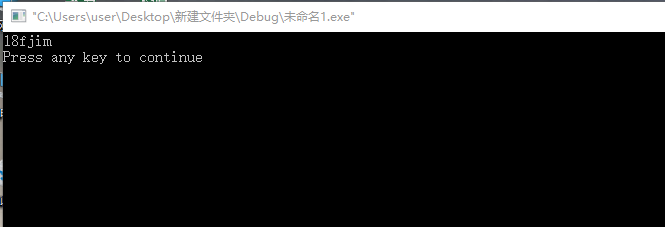


3．

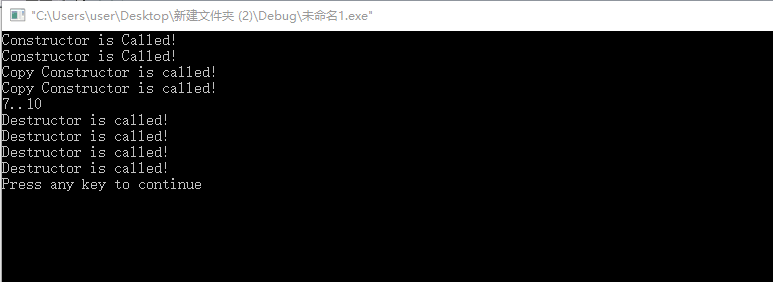
调试：程序正常，无报错



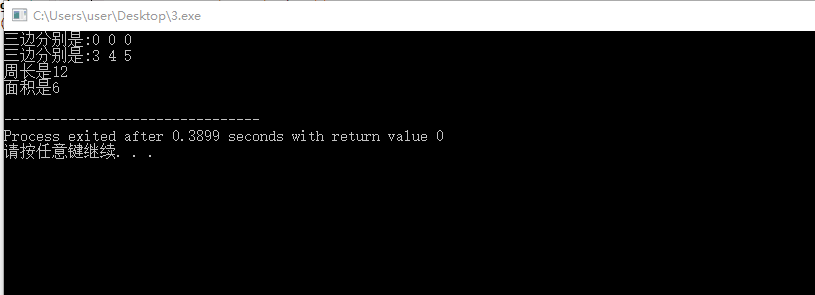
六、实验结果

1.

2.



3.



七、总结

经过这次的上机实验，我熟悉了类、类的数据成员及类的成员函数的定义方式，理解了类成员的控制方式，可以初步查看编译器的报错信息并且修改程序，并且掌握构造函数和析构函数的定义与执行过程。

**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018.10.16 指导教师签字： 刘永芬 成绩：

**实验2 类和对象（二）**

一、实验目的和要求

(1) 掌握静态成员的基本使用方法。

(2) 掌握友元的使用方法，进一步理解其概念和作用。

二、实验内容和原理

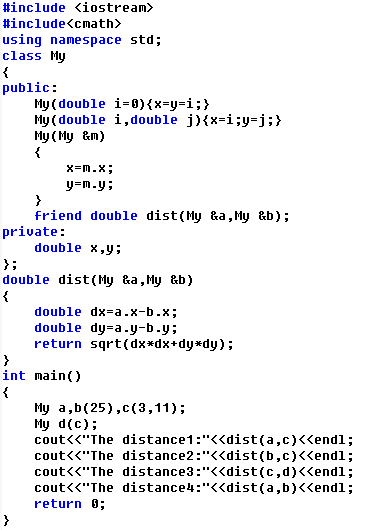
1、分析并调试程序，然后完成以下问题：

（1）指出默认参数的构造函数

（2）指出其余所有的构造函数，它们在本程序中分别起什么作用？

（3）指出友元函数。将友元函数放到私有部分，观察结果是否有变化。

（4）写出输出结果，并分析输出结果。



2、声明Circle和Rectangle两个类，Circle有半径radius属性，Rectangle有length属性和width属性，两个类只有各自的构造函数，均无成员函数，定义二者的一个友元函数compare( )用于比较圆形和矩形面积的大小，并输出。

3、定义一个Small\_Dog类，在该类定义中包括一个数据成员weight（重量），两个静态数据成员分别是total\_weight（总重量）和total\_number（总数量）；构造函数Small\_Dog(double w)用于设置重量、求总重量和总数量；成员函数Display()用于显示某只狗的重量；静态成员函数Total\_Disp()用于显示狗的总数量和总重量。在主函数main（）中，生成多个对象，调用Display()和Total\_Disp()方法，查看输出结果。

三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1.

一个参数的构造函数My(double i=0)，两个参数的构造函数My(double i,double j，友元函数friend double dist(My &a,My &b)。

#include<iostream>

#include<cmath>

using namespace std;

class My

{

public:

My(double i=0){x=y=i;}

My(double i,double j){x=i;y=j;}

My(My &m){

x=m.x;

y=m.y;

}

friend double dist(My &a,My &b);

private:

double x,y;

};

double dist(My &a,My &b)

{

double dx=a.x-b.x;

double dy=a.y-b.y;

return sqrt(dx\*dx+dy\*dy);

}

int main()

{

My a,b(25),c(3,11);

My d(c);

cout<<"The distance1:"<<dist(a,c)<<endl;

cout<<"The distance2:"<<dist(b,c)<<endl;

cout<<"The distance3:"<<dist(c,d)<<endl;

cout<<"The distance4:"<<dist(a,b)<<endl;

return 0;

}

2.

#include<stdio.h>

#include<iostream>

#define PI 3.14

using namespace std;

class Circle{

public:

Circle(double x){radius=x;}

friend double yuan(Circle &a);

private:

double radius;

};

class Rectangle{

public:

Rectangle(double a,double b){length=a;width=b;}

friend double juxing(Rectangle &);

private:

double length,width;

};

double yuan(Circle &a){

double dx=PI\*a.radius\*a.radius;

return dx;

}

double juxing(Rectangle &a){

double dy=a.length\*a.width;

return dy;

}

int main(){

double x,y,z;

cout<<"请输入圆的半径，矩形的长，宽"<<endl;

cin>>x>>y>>z;

Circle C(x);

Rectangle R(y,z);

if(yuan(C)>juxing(R))

cout<<"圆的面积大"<<endl;

else cout<<"矩形的面积大"<<endl;

return 0;

}

3.

#include<iostream>

using namespace std;

class SmallDog{

public:

void gou(double w);

static void Total\_Disp();

void Display();

static double total\_weight,total\_number;

private:

double weight;

};

double SmallDog::total\_weight=5;

double SmallDog::total\_number=5;

void SmallDog::gou(double w){

this->weight=w;

total\_weight++;

total\_number+=weight;

}

void SmallDog::Display(){

cout<<"狗的重量为:"<<weight<<endl;

}

void SmallDog::Total\_Disp(){

cout<<"total\_weight:"<<total\_weight<<endl;

cout<<"total\_number:"<<total\_number<<endl;

}

int main(){

SmallDog p1,p2;

p1.gou(6);

p1.Display();

p2.Total\_Disp();

p2.gou(7);

p2.Display();

p2.Total\_Disp();

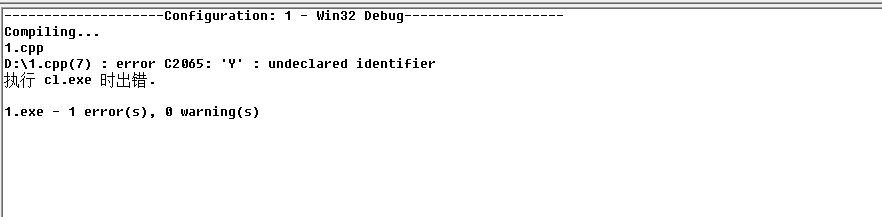
return 0;

}

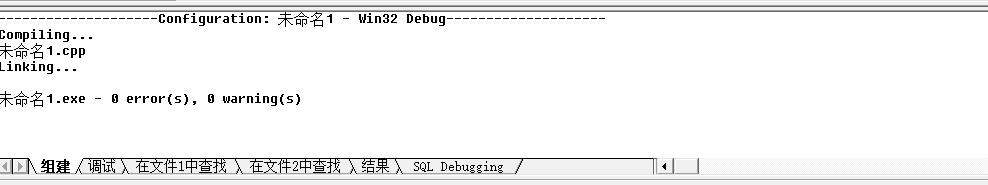
五、调试过程

1.

第一次调试错误，字母大小写错误

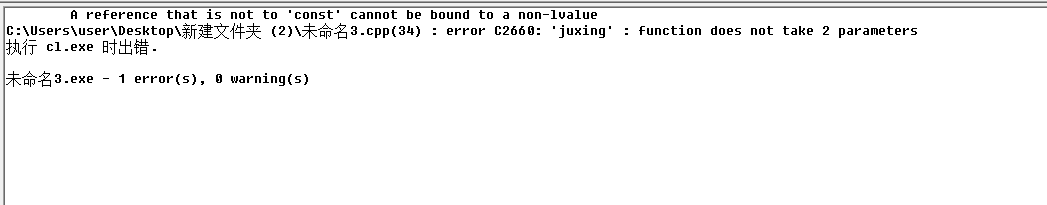


第二次调试程序运行正常

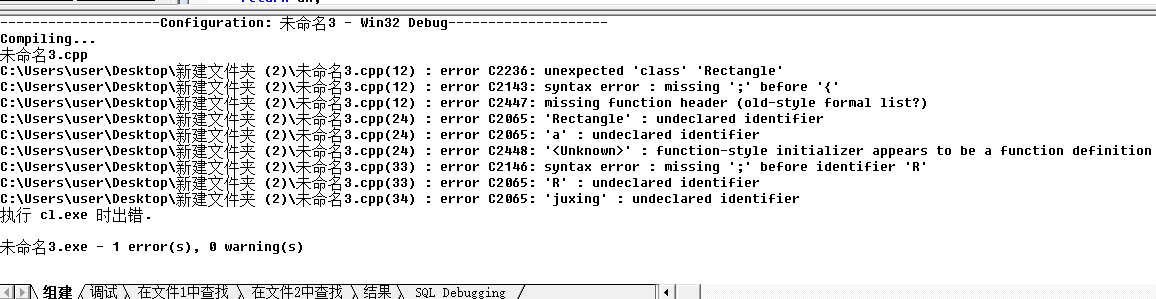


2.

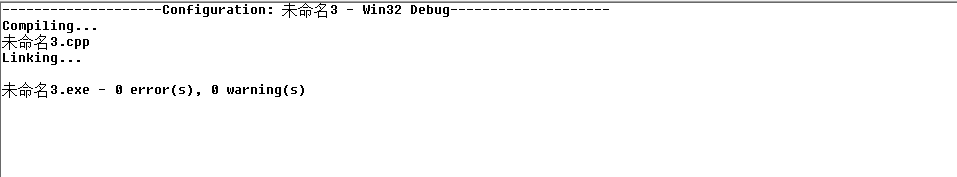
第一次调试错误：调用函数直接用半径，长宽做参数



第二次调试错误：定义类花括号后没分号

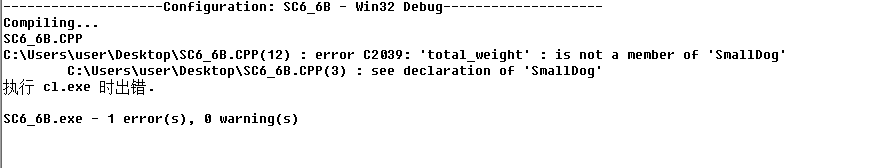


第三次调试程序正常运行

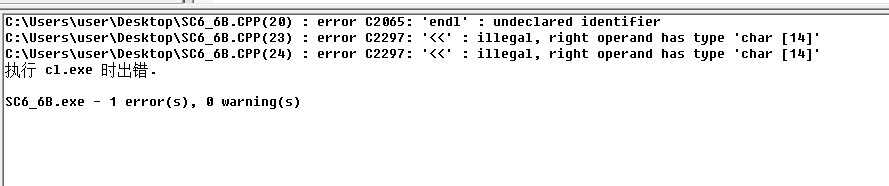


3.

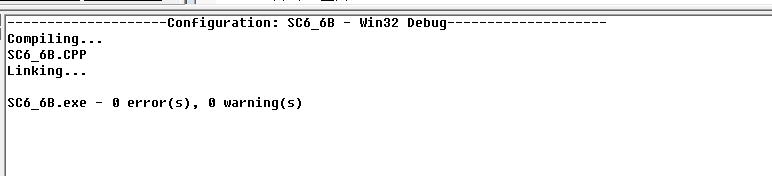
第一次调试错误：total\_weight未加下划线



第二次调试错误：未引入空间变量名

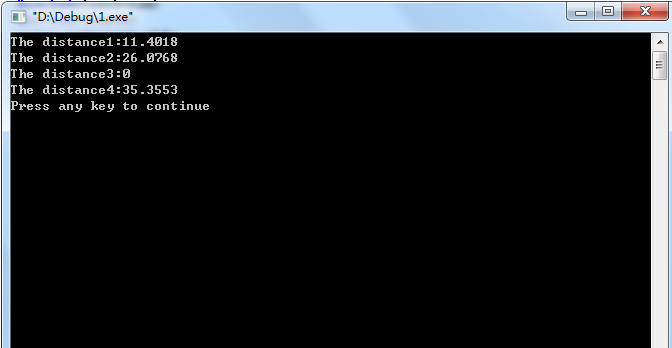


第三次调试：程序正常运行

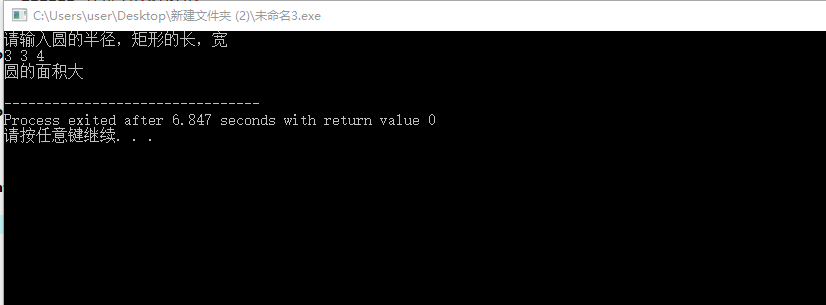


六、实验结果

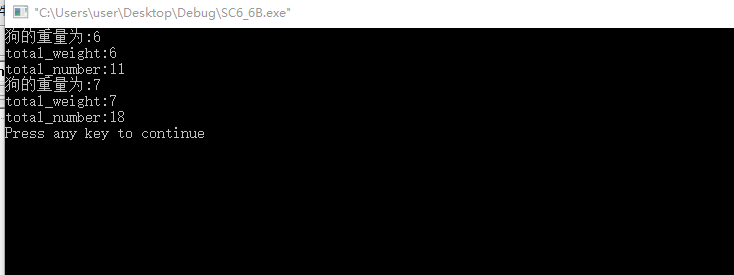
1.



2.



3.



七、总结

通过这次实验，我复习了书本上有关友元函数，静态成员函数，静态成员等知识，并能较为熟练地使用，大致理解了这些其概念和原理。通过这次实验我了解到光看书是不行的，还得动手操作才能真正掌握。

**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018.10.23 指导教师签字： 刘永芬 成绩：

**实验3 派生类和继承（一）**

一、实验目的和要求

（1）掌握派生类的声明与定义方法，进一步理解类的继承的概念，能够定义和使用类的继承关系。

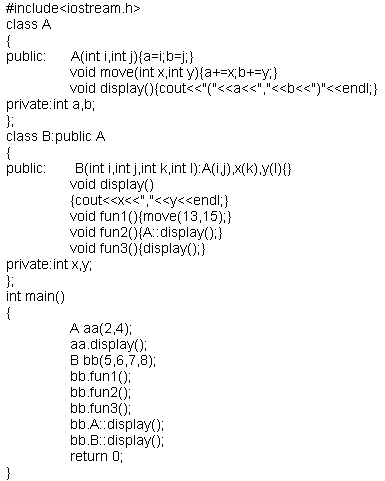
（2）熟悉公有派生和私有派生的访问特性。

二、实验内容和原理

1、(1)定义一个基类Animal，该类具有私有整型成员变量age,weight，构造派生类Dog公有继承

Animal。Dog类新增私有成员变量color，新增成员函数SetAge（int n）中直接给age赋值，新增成员函数SetWeight（int m）中直接给weight赋值，查看编译结果，并分析结果。(2)将类Animal中的age和weight为公有成员，重做第一步，并分析结果。(3)将类Animal中的age和weight为保护成员，重做第一步，并分析结果。（4）将派生类Dog的继承方式改为私有继承方式和保护继承方式重做以上各小题，并分析结果。

2、程序分析题（写出程序的输出结果，并分析结果）。



三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1.

#include<iostream>

using namespace std;

class Animal

{

public:

int age,weight;

};

class Dog :public Animal

{

private:

char color;

public:

void SetAge(int n) {age=n;}

void SetWeight(int m){weight=m;}

};

2.

#include<iostream.h>

class A

{

public: A(int i,int j){a=i;b=j;}

void move(int x,int y){a+=x;b+=y;}

void display(){cout<<"("<<a<<","<<b<<")"<<endl;}

private: int a,b;

};

class B :public A

{

public: B(int i,int j,int k,int l):A(i,j),x(k),y(l){}

void display()

{

cout<<x<<","<<y<<endl;

}

void fun1(){move(13,15);}

void fun2(){A::display();}

void fun3(){display();}

private: int x,y;

};

int main(){

A aa(2,4);

aa.display();

B bb(5,6,7,8);

bb.fun1();

bb.fun2();

bb.fun3();

bb.A::display();

bb.B::display();

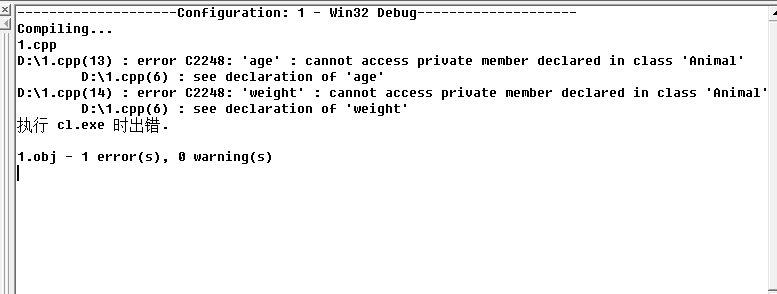
return 0;

}

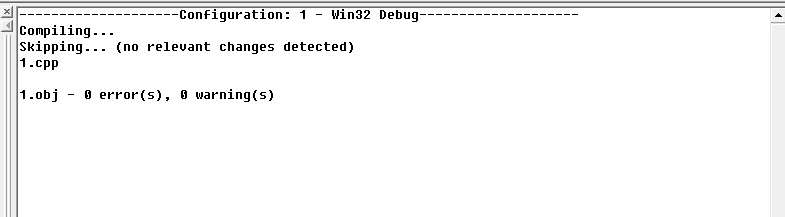
五、调试过程

1.

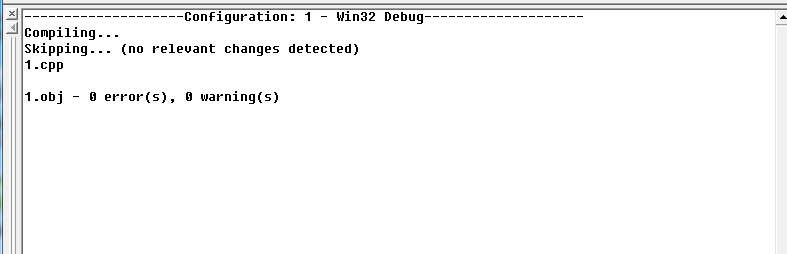
第一次调试出错：公有继承无法访问父类中私有成员



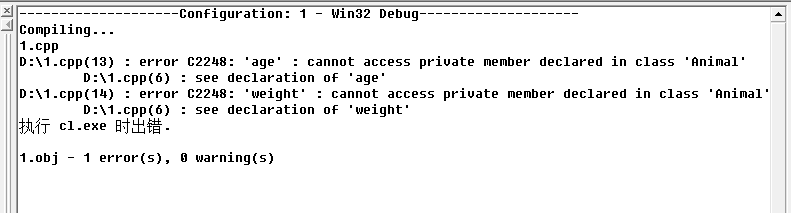
第二次调试公有继承可以访问父类中公有成员



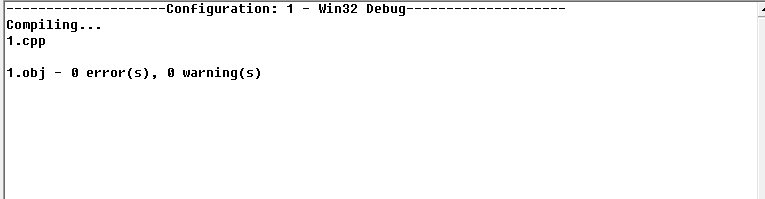
第三次调试公有继承可以访问父类中保护成员



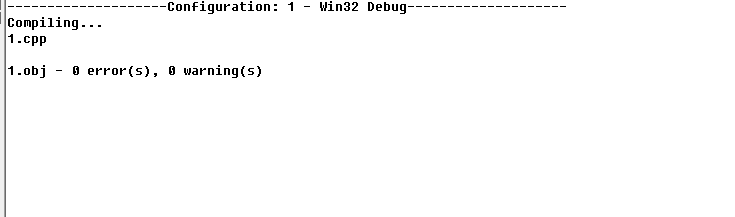
第四次调试出错：私有继承无法访问父类中私有成员



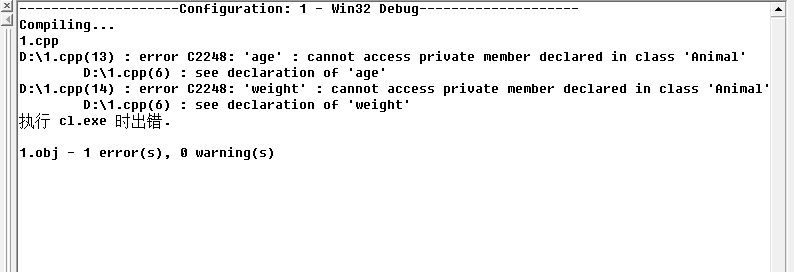
第五次调试私有继承可以访问父类中公有成员



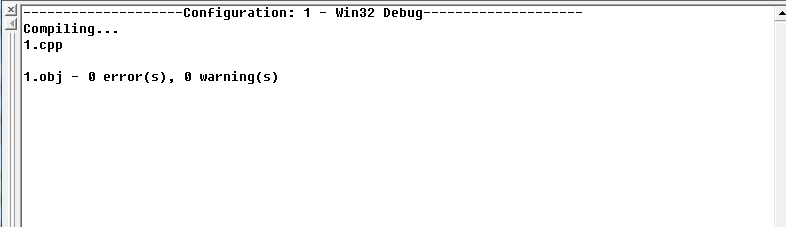
第六次调试私有继承可以访问父类中保护成员



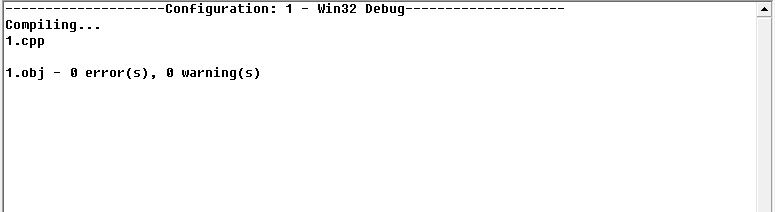
第七次调试出错：保护继承无法访问父类中私有成员



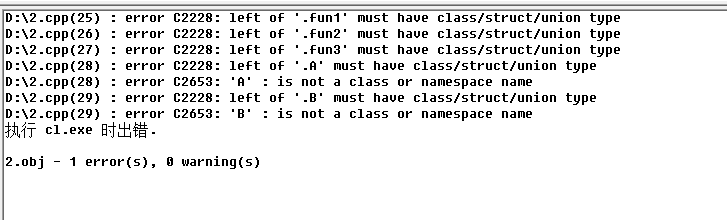
第八次调试保护继承可以访问父类中公有成员



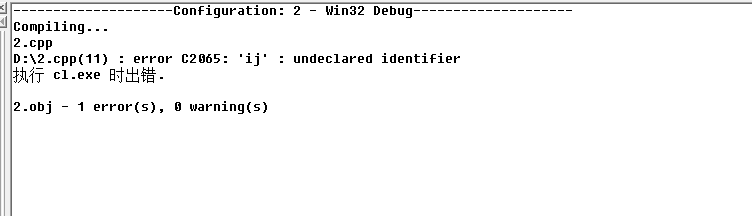
第九次调试保护继承可以访问父类中保护成员



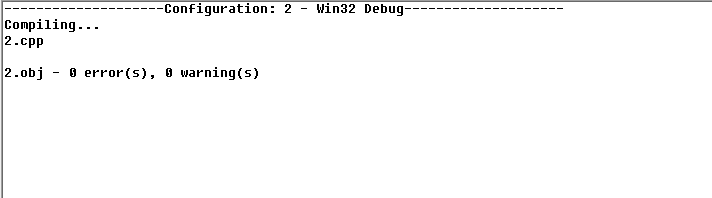
2.类B继承A时没有加分号class B :public A



类B中调用父类的函数，格式错误A(i,j)



程序运行正确



六、实验结果

1.

2.



主函数中A aa(2,4)，创建一个对象aa,并且利用构造函数赋值a=2；b=4；

aa.display()，输出对象aa中的a和b分别为2，4。

B bb(5,6,7,8)，创建一个对象bb，并且给与对象成员赋值： i=5;j=6;k=7;l=8，所以a=5 b=6 x=7 y=8。

bb.fun1()，a=5,b=6 a=5+13=18 b=6+15=21；

bb.fun2()，输出对象bb中的a和b 分别为18，21。

bb.B::display()，输出对象bb中的x和y的值7，8。

七、总结

通过这次实验我掌握了派生类的声明与定义方法，更进一步理解类的继承的概念，能够定义和使用类的继承关系，熟悉了各种继承父类的派生类对父类中各种数据类型的访问权限。

**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018.10.30 指导教师签字： 刘永芬 成绩：

**实验3 派生类和继承（二）**

一、实验目的和要求

（1）掌握派生类的声明与定义方法，进一步理解类的继承的概念，能够定义和使用类的继承关系。

（2）熟悉公有派生和私有派生的访问特性。

（3）了解虚基类在解决二义性问题中的作用。

二、实验内容和原理

1、试生成一个形状层次类结构。生成的层次结构以Shape作为基类，并由此派生出Circle类，包含成员变量半径、成员函数周长perimeter，面积area；从Circle类派生出Cylinder类，有成员变量高度，成员函数体积volume。并用函数main进行测试，输出圆对象的周长面积，圆柱对象的体积。

2、定义一个车基类vehicle，具有最大速度、重量等成员变量，设置速度、设置重量、输出启动、停止等成员函数，由车类派生出自行车类bicycle和汽车类car，自行车类具有高度等成员变量，汽车类具有座位数等成员变量，从自行车类和汽车类派生出摩托车类motorcycle，在main函数中创建一个摩托车类对象，输出它的最大速度、重量、高度、座位数。如果没有把车类定义为虚基类，会出现什么问题？为什么？

#include<iostream.h>

class vehicle

{

protected:

int MaxSpeed;

int Weight;

public:

vehicle(int s,int w){MaxSpeed=s;Weight=w;}

void SetMaxSpeed(int s){MaxSpeed=s;}

void SetWeight(int w){Weight=w;}

void Run(){cout<<"class vehicle is running!"<<endl;}

void Stop(){cout<<"class vehicle has stopped!"<<endl;}

};

class bicycle:virtual public vehicle

{

protected:

int Height;

public:

bicycle(int h,int s,int w):vehicle(s,w)

{Height=h;}

};

class car:virtual public vehicle

{

protected:

int SeatNum;

public:

car(int S,int s,int w):vehicle(s,w)

{SeatNum=S;}

};

class motorcycle:public bicycle,public car

{

public:

motorcycle(int s,int w,int h,int S):vehicle(s,w),bicycle(h,s,w),car(S,s,w){}

void Show()

{

cout<<"MaxSpeed:"<<MaxSpeed<<endl;

cout<<"Weight:"<<Weight<<endl;

cout<<"Height:"<<Height<<endl;

cout<<"SeatNum:"<<SeatNum<<endl;

}

};

void main()

{

motorcycle mc(150,200,15,3);

mc.Show();

}

三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1.

#include<iostream>

using namespace std;

double const PI=3.14;

class shape{

};

class Circle: public shape{

public:

double perimeter()

{

return 2\*PI\*r;

}

double area()

{

return PI\*r\*r;

}

Circle(double R){r=R;}

Circle(){

cout<<"请输入半径"<<endl;

cin>>r;

}

protected:

double r;

};

class Cylinder: public Circle{

public:

double volume()

{

return PI\*r\*r\*h;

}

Cylinder(double radius,double height):Circle(radius)

{

h=height;

}

Cylinder(){

cout<<"请输入高"<<endl;

cin>>h;

}

protected:

double h;

};

int main(){

Circle c;

cout<<"圆的周长为"<<c. perimeter()<<endl;

cout<<"圆的面积为"<<c.area()<<endl;

Cylinder cy1;

cout<<"圆柱的体积为："<<cy1.volume() <<endl;

return 0;

}

2

如果没有把车类定义为虚基类，编译无法通过，因为派生类访问共同基类的成员时出现二义性，编译器无法确定访问哪个成员，若有解决可以使用作用域运算符或者引入虚基类，引进虚基类，使得公共基类在它的派生类对象中只产生一个基类子对象。

#include<iostream.h>

class vehicle

{

protected:

int MaxSpeed;

int Weight;

public:

vehicle(int s,int w){MaxSpeed=s;Weight=w;}

void SetMaxSpeed(int s){MaxSpeed=s;}

void SetWeight(int w){Weight=w;}

void Run(){cout<<"class vehicle is running!"<<endl;}

void Stop(){cout<<"class vehicle has stopped!"<<endl;}

};

class bicycle:virtual public vehicle

{

protected:

int Height;

public:

bicycle(int h,int s,int w):vehicle(s,w)

{Height=h;}

};

class car:virtual public vehicle

{

protected:

int SeatNum;

public:

car(int S,int s,int w):vehicle(s,w)

{SeatNum=S;}

};

class motorcycle:public bicycle,public car

{

public:

motorcycle(int s,int w,int h,int S):vehicle(s,w),bicycle(h,s,w),car(S,s,w){}

void Show()

{

cout<<"MaxSpeed:"<<MaxSpeed<<endl;

cout<<"Weight:"<<Weight<<endl;

cout<<"Height:"<<Height<<endl;

cout<<"SeatNum:"<<SeatNum<<endl;

}

};

void main()

{

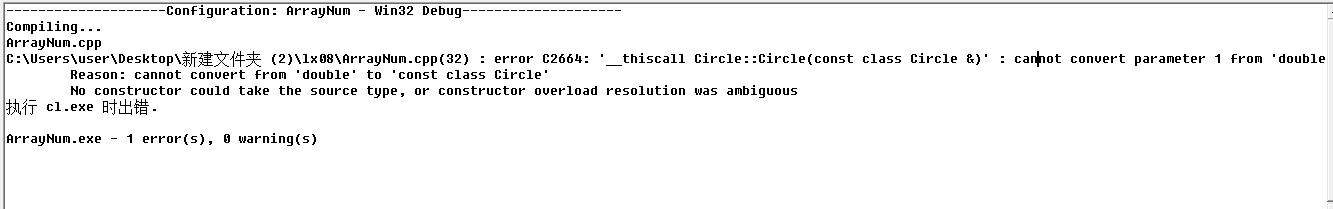
motorcycle mc(150,200,15,3);

mc.Show();

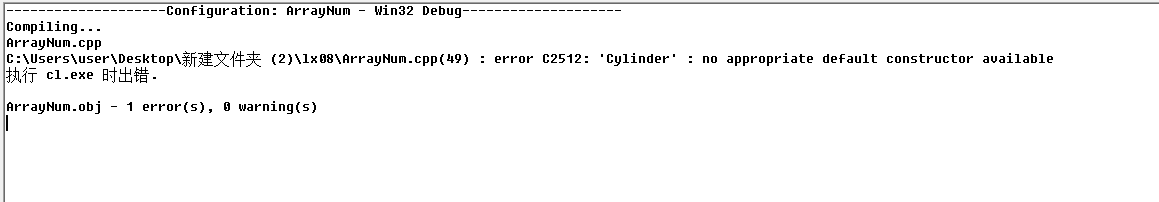
}

五、调试过程

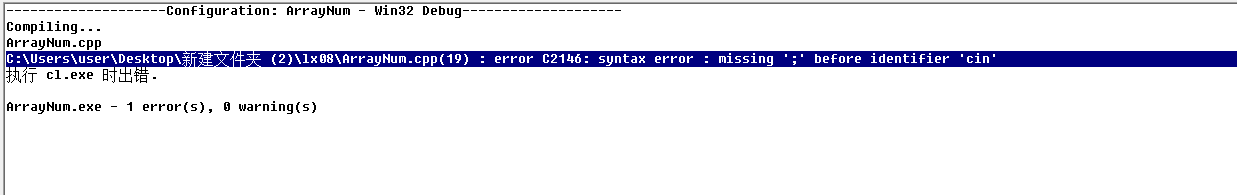
第一次调试：出错，未定义Circle的有参构造函数。



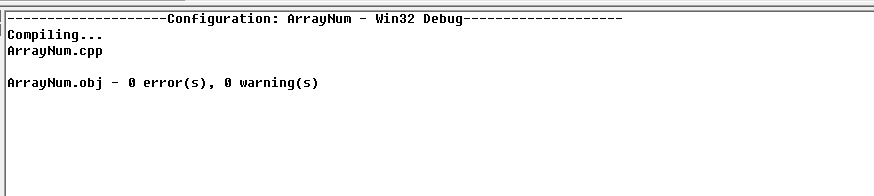
第二次调试：出错，未定义Cylinder的无参构造函数。



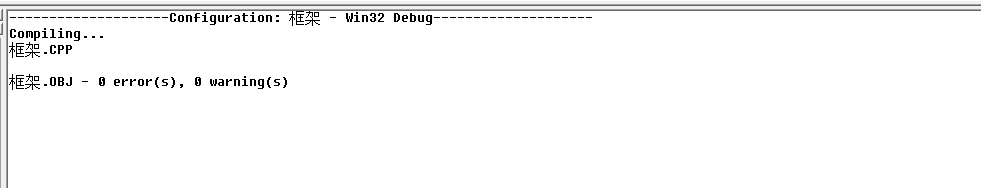
第三次调试：出错，未加分号。



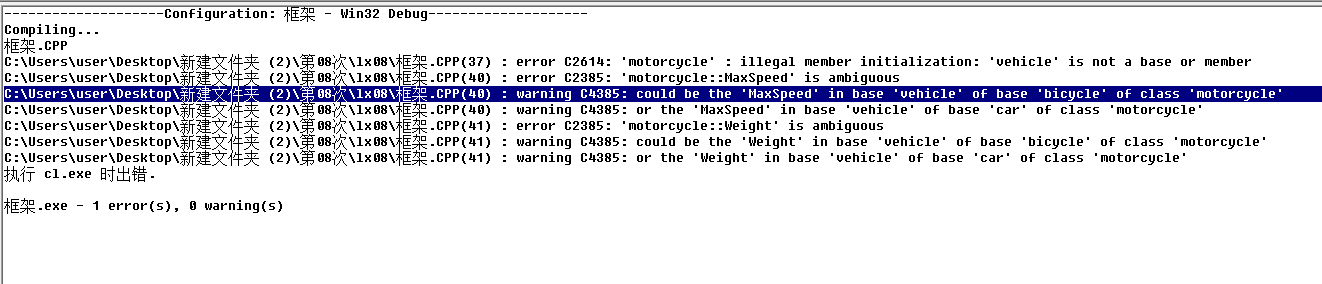
第四次调试：程序编译通过。



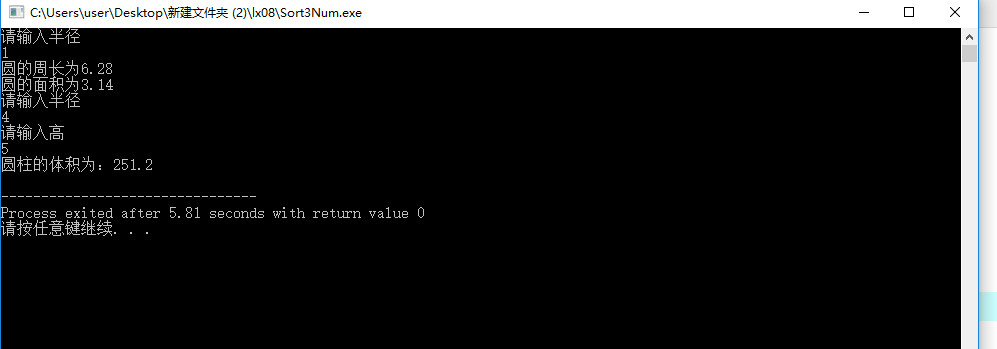
2.第一次调试程序编译通过。



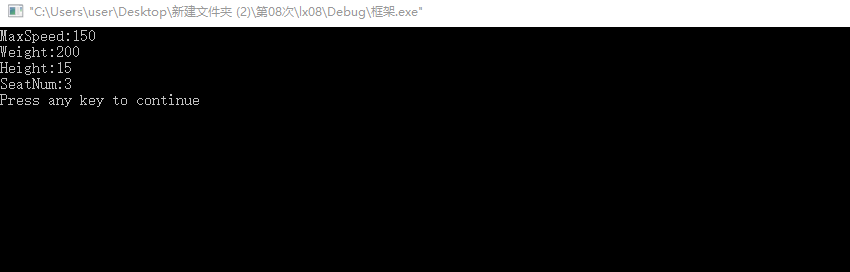
第二次调试：出错，未定义虚基类，派生类对基类成员访问的不唯一性。



六、实验结果



2.



七、总结

通过这次上机实验学习以及对书本的复习后，我掌握了派生类的声明与定义方法，进一步理解了类的继承，熟悉了公有派生和私有派生的访问特性，以及解决二义性的两种方法。认识到了理论与世间相结合的重要性，要先有理论基础才能够实践成功。

**福建农林大学金山学院实验报告**

系（教研室）：信息与机电工程系 专业： 计算机科学与技术 年级： 2017

实验课程： 面向对象程序设计 姓名： 肖伟 学号： 176709010 班级： 一班

实验室号 B704 实验时间： 2018.11.06 指导教师签字： 刘永芬 成绩：

**实验4 多态性（一）**

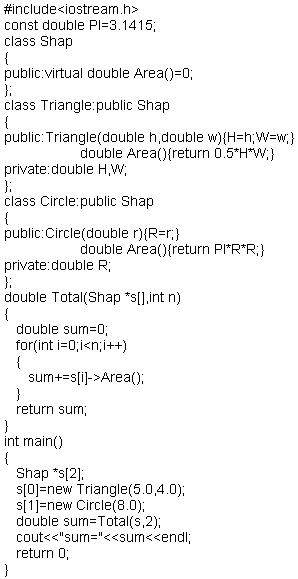
一、实验目的和要求

（1）掌握虚函数的定义与使用方法，进一步理解多态性的概念和分类。

（2）了解纯虚函数和抽象类的使用方法。

二、实验内容和原理

1、分析并调试下列程序，回答以下问题：（1）指出抽象类（2）指出虚函数，并说明它的作用（3）每个类的作用是什么？整个程序的作用是什么？



2、使用虚函数编写程序求球体V=4/3π×r³ 、圆柱体π r² h和圆锥1/3π r² h的体积，由于球体、圆柱体和圆锥都可以看做由圆继承而来，所以可以定义圆类作为基类。在圆类中定义数据成员半径和一个求体积的虚函数。由圆类派生出球体类、圆柱体类和圆锥类，在派生类中对圆类中的虚函数重新定义。编写一个外部函数求各类形状的总体积。最后在main（）函数中构造若干形状，并求它们的体积和。

三、实验环境

1. 硬件：PC机；

2. 软件：Windows操作系统、Visual C++ 6.0

四、算法描述及实验步骤

1.

抽象类是Shap。虚函数是Shap中的Area函数，它的作用是为派生类提供一个求面积的函数接口。Shap类的作用：为派生类提供一个一致的接口。Triangle类的作用是求三角形的面积。Circle类的作用是求圆形的面积。整个程序的作用是计算计算三角形和圆形的面积之和。

#include<iostream.h>

const double pi=3.1415;

class Shap

{

public:virtual double Area()=0;

};

class Triangle:public Shap

{

public:Triangle(double h,double w){H=h,W=w;}

double Area(){return 0.5\*H\*W;}

private:double H,W;

};

class Cricle:public Shap

{

public:Cricle(double r){R=r;}

double Area(){return pi\*R\*R;}

private:double R;

};

double Total(Shap\*s[],int n)

{

double sum=0;

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

{sum+=s[i]->Area();

}return sum;

}

int main()

{Shap\*s[2];

s[0]=new Triangle(5.0,4.0);

s[1]=new Cricle(8.0);

double sum=Total(s,2);

cout<<"sum="<<sum<<endl;

return 0;

}

2.

#include<iostream.h>

const double PI=3.1415;

class Circle

{

public:virtual double Volume()=0;

};

class qiu:public Circle

{

public:qiu(double r)

{

R=r;

}

double Volume()

{

return 4\*PI\*R\*R\*R/3;

}

private:double R;

};

class zhu:public Circle

{

public:

zhu(double r,double h)

{

R=r;H=h;

}

double Volume()

{

return PI\*R\*R\*H;

}

private:double R,H;

};

class zui:public Circle

{

public:zui(double r,double h)

{

R=r;H=h;

}

double Volume()

{

return PI\*R\*R\*H/3;

}

private:double R,H;

};

double Total(Circle\*s[],int n)

{

double sum=0;

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

{

sum+=s[i]->Volume();

}

return sum;

}

int main()

{

Circle \*s[3];

s[0]=new qiu(1.0);

s[1]=new zhu(1.0,2.0);

s[2]=new zui(2.0,4.0);

double sum=Total(s,3);

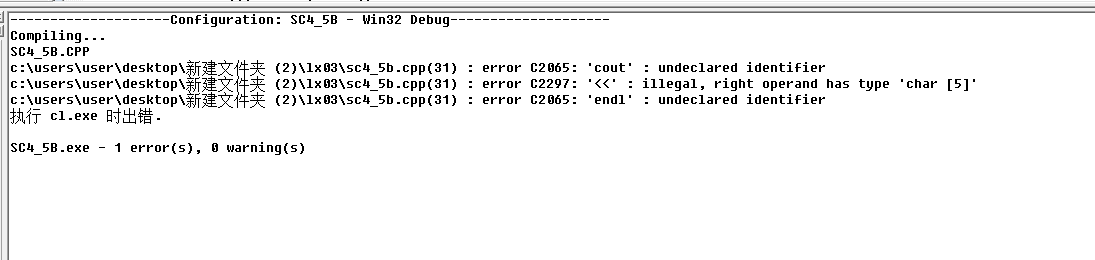
cout<<"sum="<<sum<<endl;

return 0;

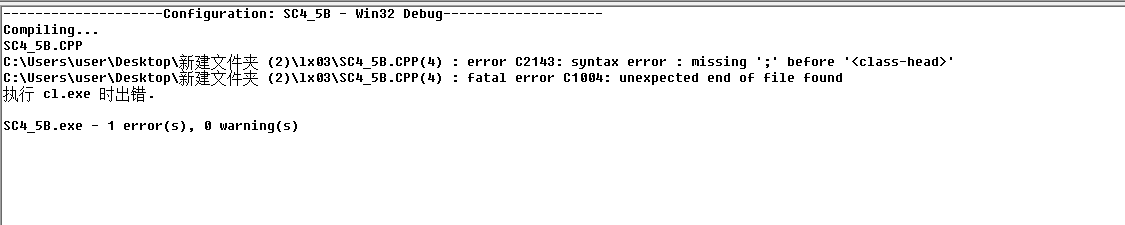
}

五、调试过程

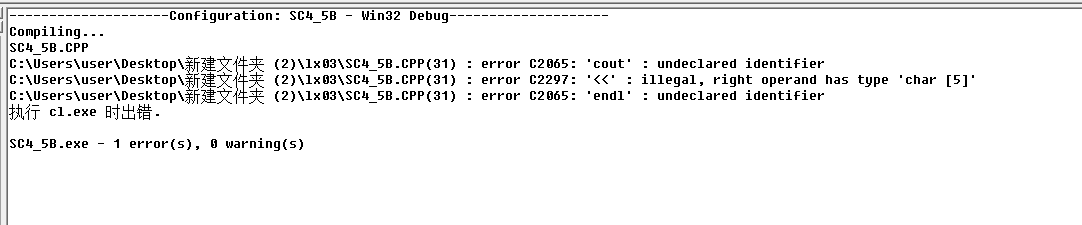
1.第一次调试报错：头文件未加.h



第二次调试报错：const 语句后未加分号

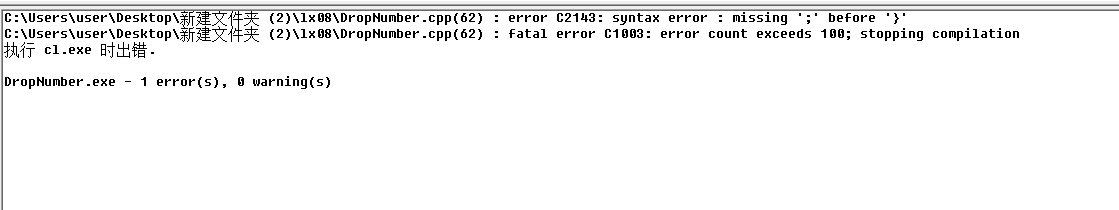


第三次调试编译通过，未报错

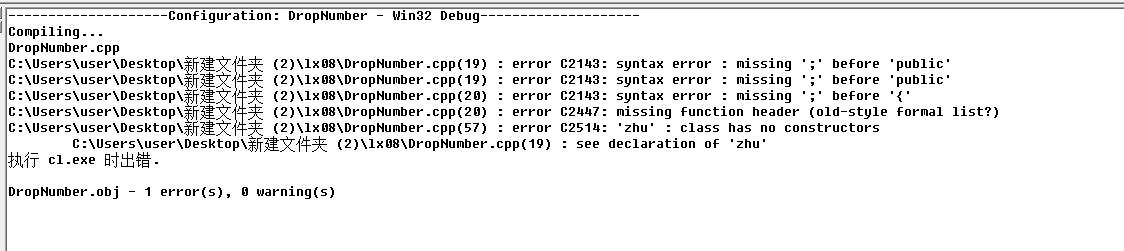


2.

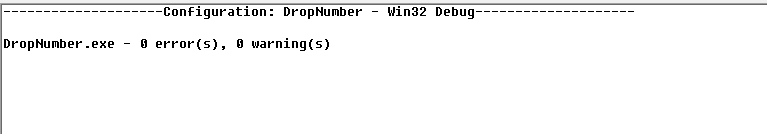
第一次调试报错：类尾未加分号



第二次调试报错：派生类继承未加分号

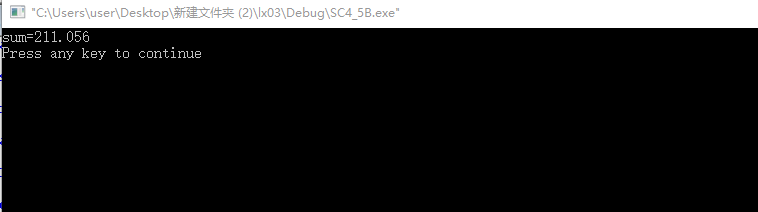


第三次调试未报错，程序正常运行

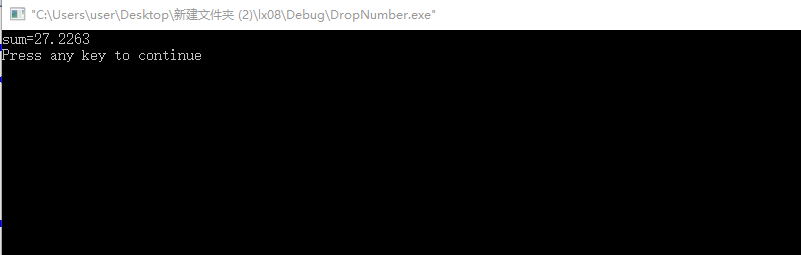


六、实验结果

1.



2.



七、总结

通过这次上机实验，我掌握了虚函数的定义与使用方法，并进一步理解多态性的概念和分类，通过看书和上机实践，大致了解了静态联编和动态联编的概念以及纯虚函数和抽象类的使用方法。