**实验名称 实验10. 多态性与虚函数 优**

1. **实验内容及结果**

写一个程序，定义抽象基类Shape，由它派生出5个派生类：Circle（圆形），Square（正方形），Rectangle（矩形），Trapezoid（梯形），Triangle（三角形）。用虚函数分别计算几种图形的面积，并求它们的和。要求用基类指针数组，使它的每一个元素指向一个派生类对象。另外，类的定义和类的实现分别放到不同的文件中。

**源代码：**

Shape.h文件

#include<iostream>

using namespace std;

class Shape{

public:

virtual float area() const {return 0;}

virtual void shapeName() const =0;

};

Circle.h文件

#include<C:\Users\Li\Desktop\test10\Shape.h>

class Circle:public Shape // Point是Shape的公用派生类

{protected:

float r;

public:

Circle(float=0);

void setCircle(float);

float getR() const {return r;}

virtual float area() const ;

virtual void shapeName() const {cout<<"Circle:";}

//friend ostream & operator<<(ostream &,const Circle &);

};

Circle.cpp文件

#include<C:\Users\Li\Desktop\test10\Circle.h>

Circle::Circle(float x)

{r=x;}

void Circle::setCircle(float x)

{r=x;}

float Circle::area() const{return 3.14159\*r\*r;}

Square.h 文件

#include<C:\Users\Li\Desktop\test10\Circle.h>

class Square:public Shape{

protected:

float a;

public:

Square(float=0);

void setSquare(float);

float getA() const {return a;}

virtual float area() const ;

virtual void shapeName() const {cout<<"Square:";}

};

Square.cpp 文件

#include<C:\Users\Li\Desktop\test10\Square.h>

Square::Square(float x)

{a=x;}

void Square::setSquare(float x)

{a=x;}

float Square::area() const{return a\*a;}

Rectangle.h文件

#include<C:\Users\Li\Desktop\test10\Square.h>

class Rectangle:public Shape

{protected:

float b,c;

public:

Rectangle(float=0,float=0);

void setRectangle(float,float);

float getB() const {return b;}

float getC() const {return c;}

virtual float area() const ;

virtual void shapeName() const {cout<<"Rectangle:";}

};

Rectangle.cpp文件

#include<C:\Users\Li\Desktop\test10\Rectangle.h>

Rectangle::Rectangle(float x,float y)

{b=x;c=y;}

void Rectangle::setRectangle(float x,float y)

{b=x;c=y;}

float Rectangle::area() const{return b\*c;}

Trapezoid.h文件

#include<C:\Users\Li\Desktop\test10\Rectangle.h>

class Trapezoid:public Shape

{protected:

float d,e,f;

public:

Trapezoid(float=0,float=0,float=0);

void setTrapezoid(float,float,float);

float getD() const {return d;}

float getE() const {return e;}

float getF() const {return f;}

virtual float area() const ;

virtual void shapeName() const {cout<<"Trapezoid:";}

};

Trapezoid.cpp文件

#include<C:\Users\Li\Desktop\test10\Trapezoid.h>

Trapezoid::Trapezoid(float x,float y,float z)

{d=x;e=y;f=z;}

void Trapezoid::setTrapezoid(float x,float y,float z)

{d=x;e=y;f=z;}

float Trapezoid::area() const{return (e+d)\*f/2;}

Triangle.h文件

#include<C:\Users\Li\Desktop\test10\Trapezoid.h>

class Triangle:public Shape

{protected:

float g,h;

public:

Triangle(float=0,float=0);

void settangle(float,float);

float getG() const {return g;}

float getH() const {return h;}

virtual float area() const ;

virtual void shapeName() const {cout<<"Triangle:";}

};

Triangle.cpp文件

#include<C:\Users\Li\Desktop\test10\Triangle.h>

Triangle::Triangle(float x,float y)

{g=x;h=y;}

void Triangle::settangle(float x,float y)

{g=x;h=y;}

float Triangle::area() const{return g\*h/2;}

main.cpp文件

#include<C:\Users\Li\Desktop\test10\Triangle.h>

void fun(Shape \*);

int main(){

Circle Ci(2.3);

Square Sq(4);

Rectangle Re(8.6,9.4);

Trapezoid Tr(4,6,7);

Triangle Tri(5.2,7);

Shape \*pt[5];

pt[0]=&Ci;

pt[0]->shapeName();

cout<<"半径："<<Ci.getR()<<"圆面积="<<pt[0]->area() <<"\n\n";

pt[1]=&Sq;

pt[1]->shapeName();

cout<<"边长："<<Sq.getA()<<"正方形面积："<<pt[1]->area() <<"\n\n";

pt[2]=&Re;

pt[2]->shapeName();

cout<<"长："<<Re.getB()<<"宽："<<Re.getC()<<"\n"<<"长方形面积="<<pt[2]->area() <<"\n\n";

pt[3]=&Tr;

pt[3]->shapeName();

cout<<"上底："<<Tr.getD()<<"下底："<<Tr.getE()<<"高："<<Tr.getF()<<"\n"<<"梯形面积="<<pt[3]->area() <<"\n\n";

pt[4]=&Tri;

pt[4]->shapeName();

cout<<"边长："<<Tri.getG()<<"高："<<Tri.getH()<<"\n"<<"三角形面积="<<pt[4]->area() <<"\n\n";

int i;

float sum=0;

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

sum=sum+pt[i]->area();

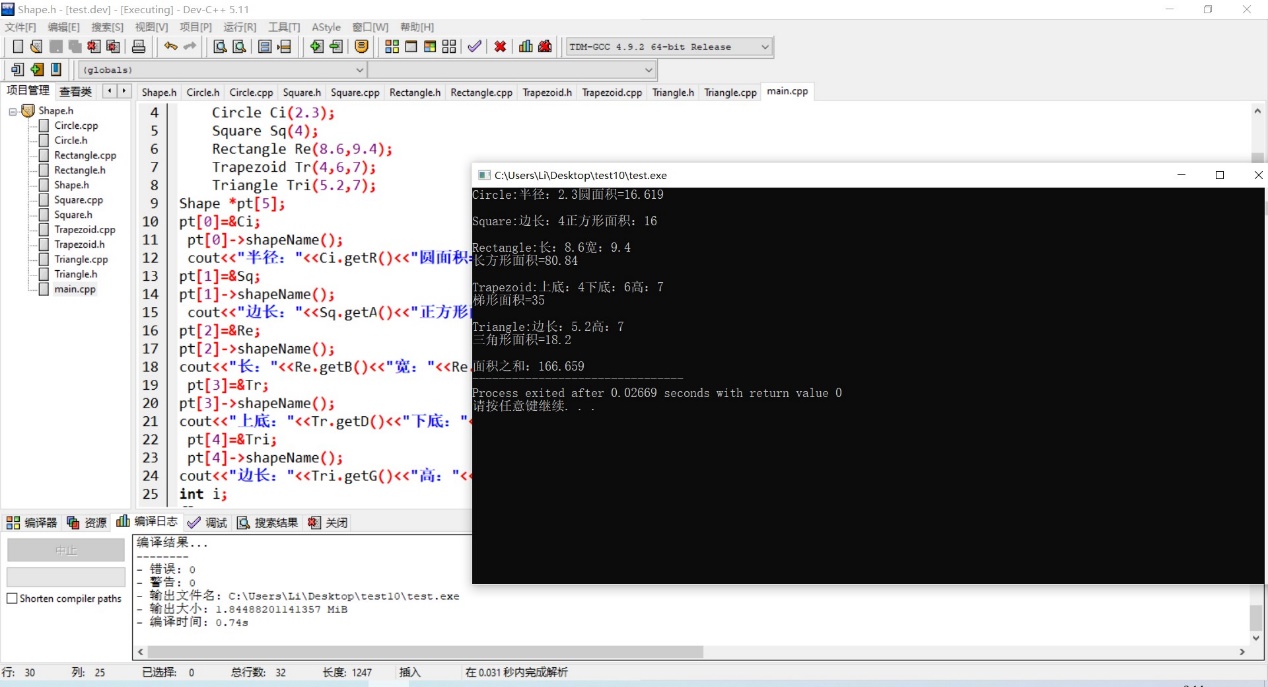
}

cout<<"面积之和："<<sum;

return 0;

}

**运行结果（截图）**：



1. **实验体会**

**第一次敲这么长的代码，但是总体难度不大。**

**非常好！**