## 实验十

学号： 姓名： 专业：

**1 实验目的：掌握面向对象编程中继承、派生的用法。**

**2 实验题目：**

**（1）编写一个集合类，要求类的每个对象可以保存0~100个不同的整数，并使用友元函数实现如下功能：**

①从集合中加入一个整数

②从集合中去掉一个整数

③判断一个整数是否在集合中

④求两个集合的并集，结果是一个集合

⑤求两个集合的交集，结果是一个集合

**（2）下面是一个形状类Shape，编写Shape的派生类：圆类Circle、三角形类Triangle和矩形类Rectangle，并重定义基类的成员函数使之返回正确的结果（show函数要输出对象的基本信息），然后编写程序进行测试。**

class Shape

{public:

double area()

{return 0;};

double girth()

{return 0;};

void show()

{cout<<”Shape Object”<<endl;};

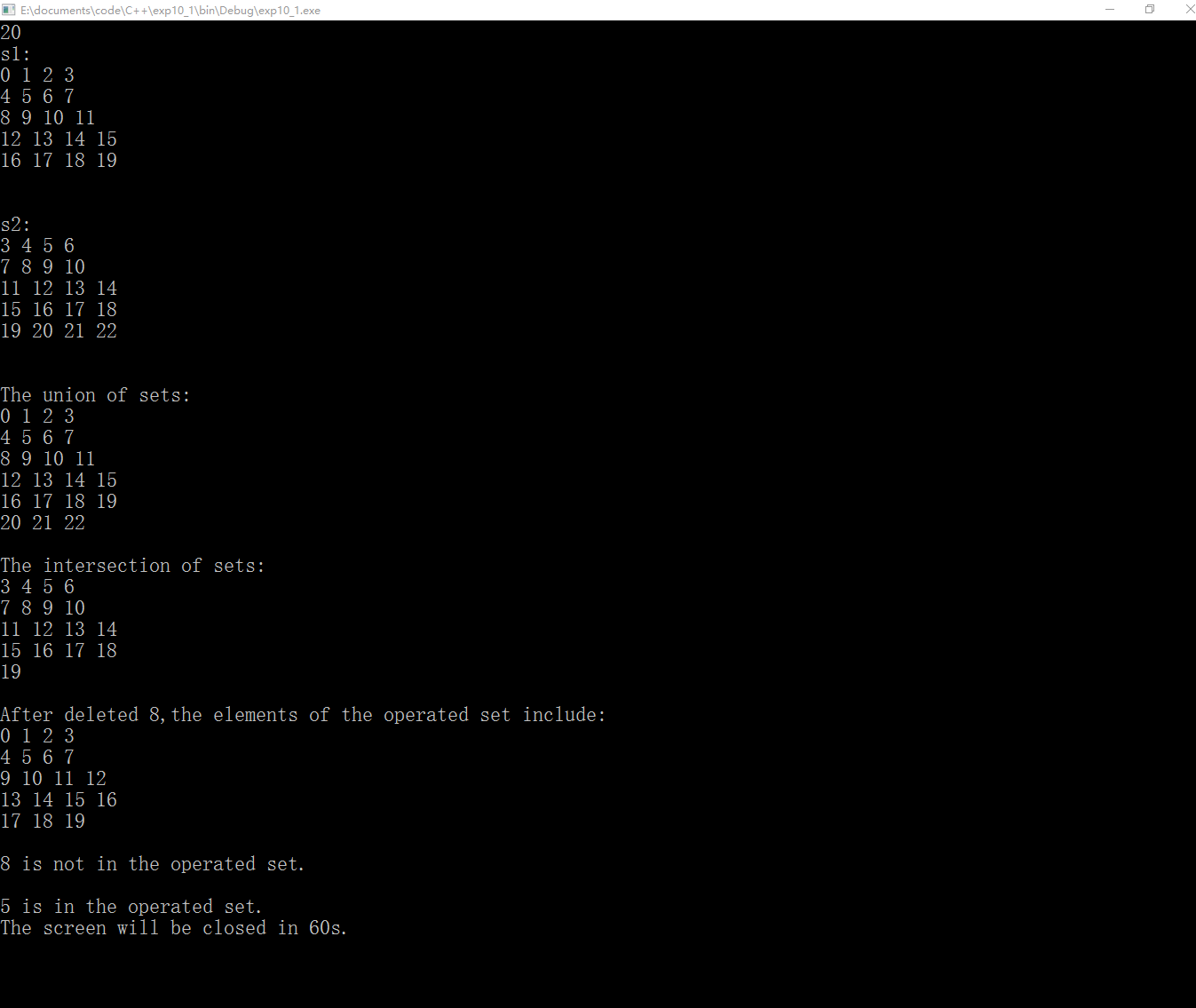
private:

}；

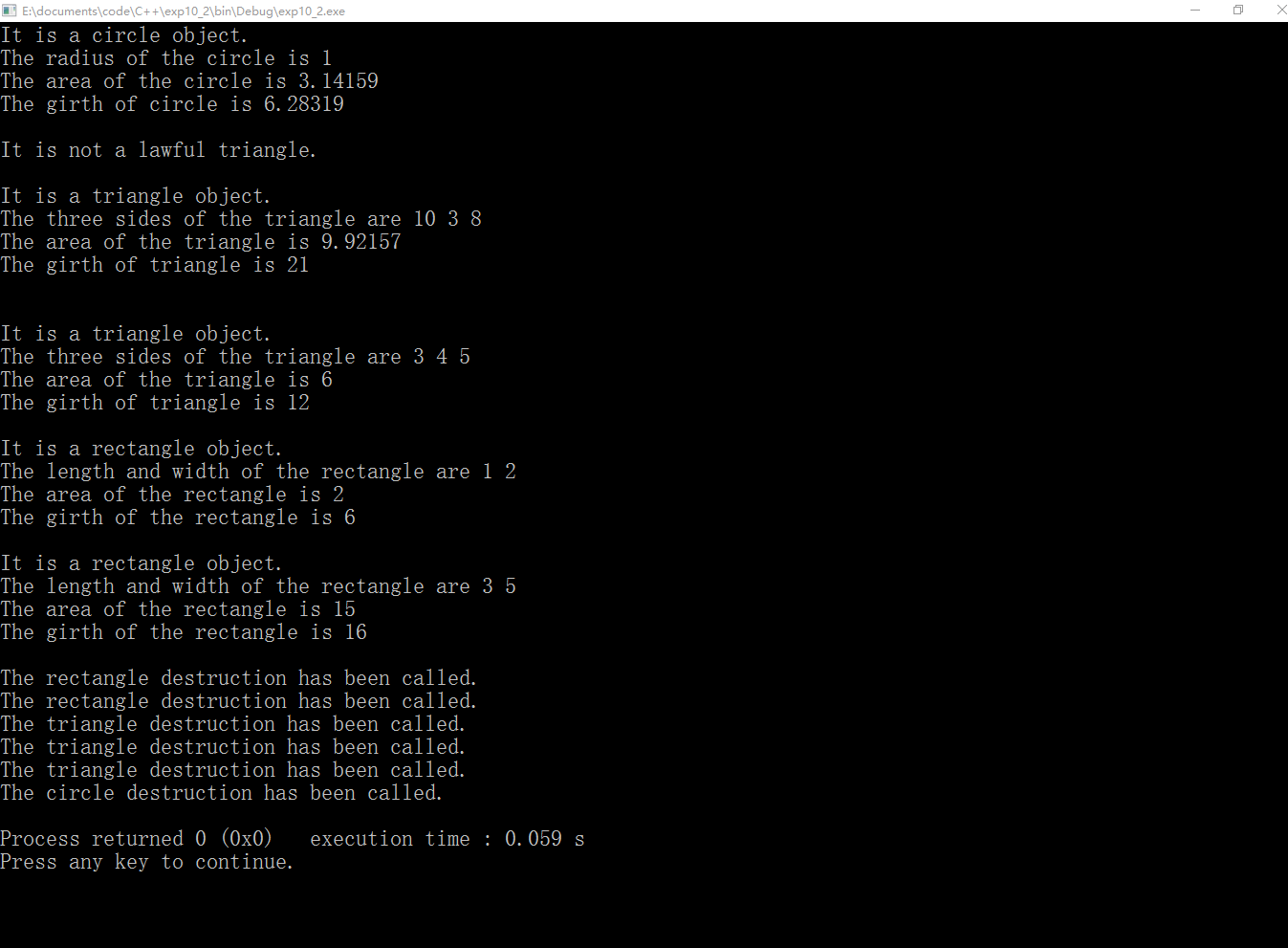
**（3）编写一个笔类，描述所有笔的共同属性，然后编写笔类的派生类：钢笔、铅笔、签字笔、毛笔，在各派生类中尽量描述清楚各自的属性。**

**（4）自学完成：定义复数类，重载“+”“-”“=”运算符。**

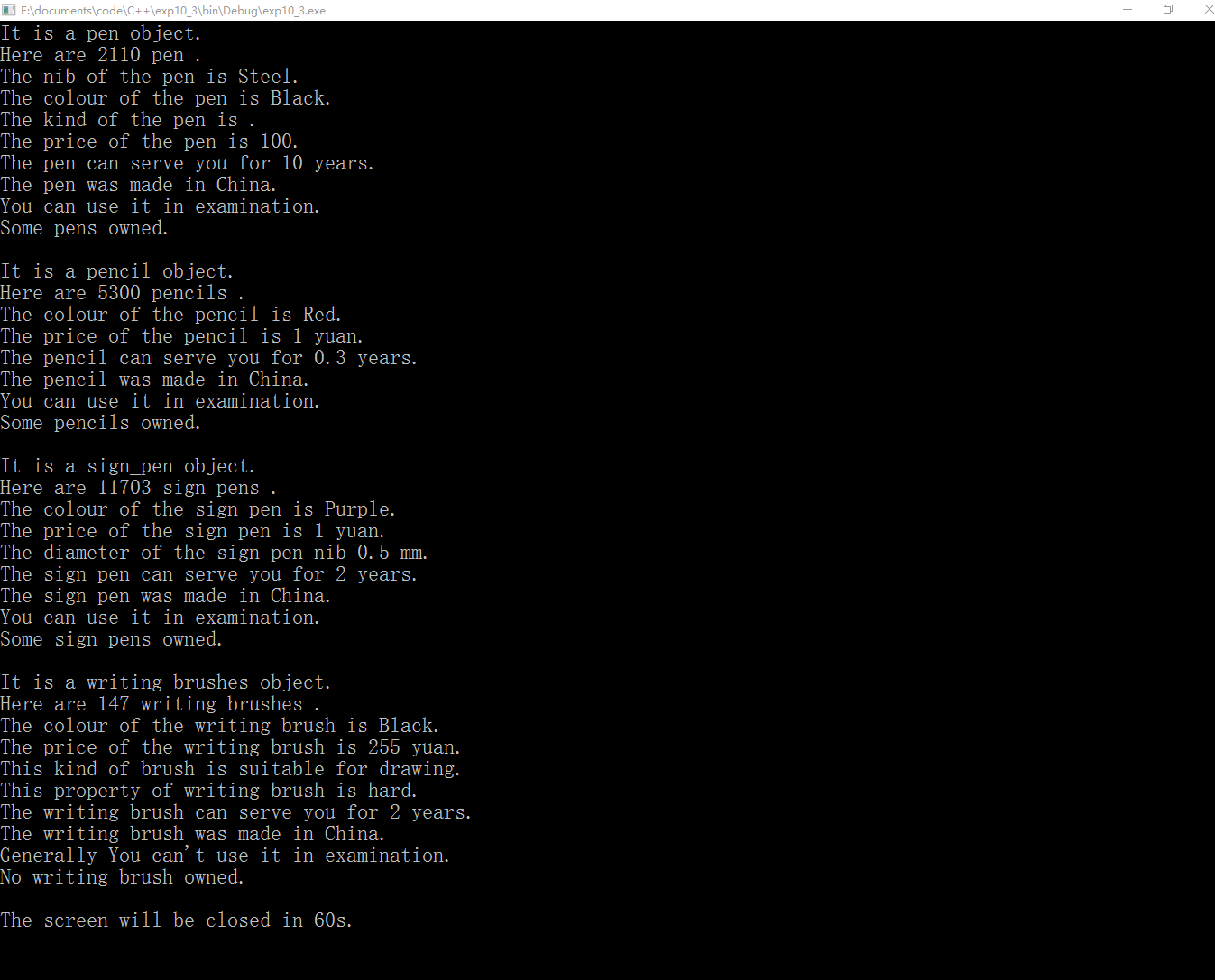
**3实验结果截图**



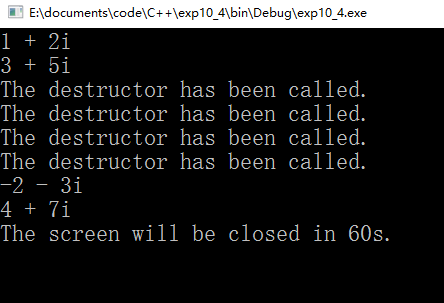
图片1题目（1）程序运行截图



图片2题目（2）程序运行截图



图片3题目（3）程序运行截图



图片4题目（4）程序运行截图

**4源代码**

//题目（1）源代码：

//\_set.h

#ifndef \_SET\_H\_INCLUDED

#define \_SET\_H\_INCLUDED

#include <iostream>

#include <cstdio>

#include <windows.h>

class \_set

**{**

friend \_add**(**\_set **&,**int**);**

friend \_del**(**\_set **&,**int**);**

friendbool \_judge**(**\_set **&,**int**);**

//friend \_set &\_union(\_set &,\_set &);//求并集

//friend \_set &\_intersection(\_set &,\_set &);

friend \_set \_union**(**\_set **&,**\_set **&);**//求并集

friend \_set \_intersection**(**\_set **&,**\_set **&);**//求交集

public**:**

\_set**(**int**);**

\_set**();**

**~**\_set**();**

display**(**\_set **&);**

private**:**

int**\***p**;**

int \_size**;**//size of number

**};**

#endif // \_SET\_H\_INCLUDED

// \_set.cpp: implementation of the \_set class.

//

//////////////////////////////////////////////////////////////////////

#include "\_set.h"

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

\_set**::**\_set**()**

**{**

**}**

\_add**(**\_set **&**s**,**int n**)**

**{**

s**.**p**[**s**.**\_size**++]=**n**;**

**}**

\_del**(**\_set **&**s**,**int n**)**

**{**

printf**(**"After deleted %d,the elements of the operated set include:\n"**,**n**);**

int index**;**

**for(**inti**=**0**;**i**<**s**.**\_size**;**i**++)**

**if(**s**.**p**[**i**]==**n**)**

index**=**i**;**

**for(**int j**=**index**;**j**<**s**.**\_size**;**j**++)**

**{**

s**.**p**[**j**]=**s**.**p**[**j**+**1**];**

**}**

s**.**\_size**--;**

**}**

bool \_judge**(**\_set **&**s**,**int n**)**

**{**

**for(**inti**=**0**;**i**<**s**.**\_size**;**i**++)**

**if(**s**.**p**[**i**]==**n**)**

**{**

printf**(**"%d is in the operated set.\n"**,**n**);**

**returntrue;**

**}**

printf**(**"%d is not in the operated set.\n"**,**n**);**

**returnfalse;**

**}**

\_set \_union**(**\_set **&**s1**,**\_set **&**s2**)**//求并集

//\_set \_union(\_set &s1,\_set &s2)//求并集

**{**

\_set s3**(**s1**.**\_size**+**s2**.**\_size**);**

**for(**inti**=**0**;**i**<**s1**.**\_size**;**i**++)**

s3**.**p**[**s3**.**\_size**++]=**s1**.**p**[**i**];**

**for(**int j**=**0**;**j**<**s2**.**\_size**;**j**++)**

**{**

int index**=**0**;**

**for(;**index**<**s2**.**\_size**;**index**++)**

**if(**s3**.**p**[**index**]==**s2**.**p**[**j**])**

//continue;

**break;**

**if(**index**==**s1**.**\_size**)**

**{**

s3**.**p**[**s3**.**\_size**++]=**s2**.**p**[**j**];**

**}**

**}**

std**::**cout**<<**"The union of sets:"**<<**std**::**endl**;**

**return** s3**;**

**}**

\_set \_intersection**(**\_set **&**s1**,**\_set **&**s2**)**

//\_set \_intersection(\_set &s1,\_set &s2)

**{**

\_set s3**(**s1**.**\_size**+**s2**.**\_size**);**

int bigger**;**

**if(**s1**.**\_size**>**s2**.**\_size**)**

bigger**=**s1**.**\_size**;**

**else**

bigger**=**s2**.**\_size**;**

**for(**inti**=**0**;**i**<**bigger**;**i**++)**

**{**

**for(**int j**=**0**;**j**<**bigger**;**j**++)**

**{**

**if(**s1**.**p**[**i**]==**s2**.**p**[**j**])**

s3**.**p**[**s3**.**\_size**++]=**s1**.**p**[**i**];**

**}**

**}**

std**::**cout**<<**"The intersection of sets:"**<<**std**::**endl**;**

**return** s3**;**

**}**

\_set**::~**\_set**()**

**{**

**delete[]**p**;**

std**::**cout**<<**"The destruction has been called"**<<**std**::**endl**;**

**}**

\_set**::**\_set**(**int n**)**

**{**

p**=new**int**[**n**];**

\_size**=**0**;**

**}**

\_set**::**display**(**\_set **&**s**)**

**{**

**for(**inti**=**0**;**i**<**s**.**\_size**;**i**++)**

**{**

std**::**cout**<<**s**.**p**[**i**]<<**" "**;**

**if((**i**+**1**)%**4**==**0**)**

std**::**cout**<<**std**::**endl**;**

**}**

**}**

//test.cpp

#include "\_set.h"

int main**()**//测了几次感觉没毛病

**{**

int n**;**

std**::**cin**>>**n**;**

\_set s1**(**n**),**s2**(**n**);**

**for(**inti**=**0**;**i**<**n**;**i**++)**

**{**

\_add**(**s1**,**i**);**

\_add**(**s2**,**i**+**3**);**

**}**

std**::**cout**<<**"s1:\n"**;**

s1**.**display**(**s1**);**

std**::**cout**<<**std**::**endl**;**

std**::**cout**<<**std**::**endl**;**

std**::**cout**<<**"s2:\n"**;**

s2**.**display**(**s2**);**

std**::**cout**<<**"\n\n"**;**

\_set s3**=**\_union**(**s1**,**s2**);**

s3**.**display**(**s3**);**

std**::**cout**<<**"\n\n"**;**

\_set s4**=**\_intersection**(**s1**,**s2**);**

s4**.**display**(**s4**);**

std**::**cout**<<**"\n\n"**;**

\_del**(**s1**,**8**);**

s1**.**display**(**s1**);**

std**::**cout**<<**"\n\n"**;**

\_judge**(**s1**,**8**);**

std**::**cout**<<**"\n"**;**

\_judge**(**s2**,**5**);**

std**::**cout**<<**"The screen will be closed in 60s."**<<**std**::**endl**;**

Sleep**(**1000**\***60**);**

**return**0**;**

**}**

//题目（2）源代码：

//Shape.h

#ifndef SHAPE\_H\_INCLUDED

#define SHAPE\_H\_INCLUDED

#include <iostream>

//#include <windows.h>

class Shape

**{**

public**:**

Shape**();**

virtual**~**Shape**();**

doublearea**(){return**0**;};**

doublegirth**(){return**0**;};**

virtualvoidshow**(){** std**::**cout**<<**"Shape Object"**<<**std**::**endl**;};**

protected**:**

double \_girth**,**\_area**;**

**};**

#endif // SHAPE\_H\_INCLUDED

// Shape.cpp: implementation of the Shape class.

//

//////////////////////////////////////////////////////////////////////

#include "Shape.h"

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

Shape**::**Shape**()**

**{**

**}**

Shape**::~**Shape**()**

**{**

**}**

//Circle.h

#ifndef CIRCLE\_H\_INCLUDED

#define CIRCLE\_H\_INCLUDED

#include "Shape.h"

#define pi 3.1415926

classCircle**:**public Shape

**{**

public**:**

Circle**();**

Circle**(**double**);**

virtual**~**Circle**();**

doublearea**();**

doublegirth**();**

voidshow**();**

private**:**

double r**;**

**};**

#endif // CIRCLE\_H\_INCLUDED

// ircle.cpp: implementation of the Circle class.

//

//////////////////////////////////////////////////////////////////////

#include "Circle.h"

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

Circle**::**Circle**()**

**{**

**}**

Circle**::**Circle**(**double R**)**

**{**

r**=**R**;**

**}**

Circle**::~**Circle**()**

**{**

std**::**cout**<<**"The circle destruction has been called."**<<**std**::**endl**;**

**}**

doubleCircle**::**area**()**

**{**

//\_area=pi\*r\*r;

**return** pi**\***r**\***r**;**

**}**

voidCircle**::**show**()**

**{**

\_area**=**area**();**

\_girth**=**girth**();**

std**::**cout**<<**"It is a circle object."**<<**std**::**endl**;**

std**::**cout**<<**"The radius of the circle is "**<<**r**<<**std**::**endl**;**

std**::**cout**<<**"The area of the circle is "**<<**\_area**<<**std**::**endl**;**

std**::**cout**<<**"The girth of circle is "**<<**\_girth**<<**std**::**endl**;**

**}**

doubleCircle**::**girth**()**

**{**

//\_girth=2\*pi\*r;

**return**2**\***pi**\***r**;**

**}**

//Rectangle.h

#ifndef RECTANGLE\_H\_INCLUDED

#define RECTANGLE\_H\_INCLUDED

//#include <windows.h>

#include "Shape.h"

classRectangle**:**public Shape

**{**

public**:**

Rectangle**();**

Rectangle**(**double**,**double**);**

**~**Rectangle**();**

doublearea**();**

doublegirth**();**

voidshow**();**

private**:**

doublelength**,**width**;**

**};**

#endif // RECTANGLE\_H\_INCLUDED

//Rectangle.cpp

#include "Rectangle.h"

Rectangle**::**Rectangle**()**

**{**

**}**

Rectangle**::**Rectangle**(**double \_a**,**double \_b**)**

**{**

length**=**\_a**;**

width**=**\_b**;**

**}**

Rectangle**::~**Rectangle**()**

**{**

std**::**cout**<<**"The rectangle destruction has been called."**<<**std**::**endl**;**

**}**

void Rectangle**::**show**()**

**{**

\_girth**=**girth**();**

\_area**=**area**();**

std**::**cout**<<**"It is a rectangle object."**<<**std**::**endl**;**

std**::**cout**<<**"The length and width of the rectangle are "**<<**length**<<**" "**<<**width**<<**std**::**endl**;**

std**::**cout**<<**"The area of the rectangle is "**<<**\_area**<<**std**::**endl**;**

std**::**cout**<<**"The girth of the rectangle is "**<<**\_girth**<<**std**::**endl**;**

**}**

double Rectangle**::**area**()**

**{**

**return** length**\***width**;**

**}**

doubleRectangle**::**girth**()**

**{**

**return**2**\*(**length**+**width**);**

**}**

//Triangle.h

#ifndef TRIANGLE\_H\_INCLUDED

#define TRIANGLE\_H\_INCLUDED

#include "Shape.h"

#include <cmath>

//#include <windows.h>

classTriangle**:**public Shape

**{**

public**:**

Triangle**();**

Triangle**(**double**,**double**,**double**);**

virtual**~**Triangle**();**

doublearea**();**

doublegirth**();**

voidshow**();**

booljudge**();**

private**:**

double \_side1**,**\_side2**,**\_side3**;**

**};**

#endif // TRIANGLE\_H\_INCLUDED

// Triangle.cpp: implementation of the Triangle class.

//

//////////////////////////////////////////////////////////////////////

#include "Triangle.h"

#include <math.h>

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

Triangle**::**Triangle**()**

**{**

**}**

Triangle**::**Triangle**(**double \_a**,**double \_b**,**double \_c**)**

**{**

\_side1**=**\_a**;**

\_side2**=**\_b**;**

\_side3**=**\_c**;**

**}**

Triangle**::~**Triangle**()**

**{**

std**::**cout**<<**"The triangle destruction has been called."**<<**std**::**endl**;**

**}**

double Triangle**::**area**()**

**{**

**if(**judge**())**

**{**

double s**=**0.0**,**l**=(**\_side1**+**\_side2**+**\_side3**)/**2**;**

s**=**sqrt**(**l**\*(**l**-**\_side1**)\*(**l**-**\_side2**)\*(**l**-**\_side3**));**

**return** s**;**

**}**

**return**0.0**;**

**}**

double Triangle**::**girth**()**

**{**

**if(**judge**())**

**return** \_side1**+**\_side2**+**\_side3**;**

**return**0.0**;**

**}**

void Triangle**::**show**()**

**{**

**if(**judge**())**

**{**

\_area**=**area**();**

\_girth**=**girth**();**

std**::**cout**<<**"It is a triangle object."**<<**std**::**endl**;**

std**::**cout**<<**"The three sides of the triangle are "**<<**\_side1**<<**" "**<<**\_side2**<<**" "**<<**\_side3**<<**" "**<<**std**::**endl**;**

std**::**cout**<<**"The area of the triangle is "**<<**\_area**<<**std**::**endl**;**

std**::**cout**<<**"The girth of triangle is "**<<**\_girth**<<**std**::**endl**;**

**}**

**else**

std**::**cout**<<**"It is not a lawful triangle."**<<**std**::**endl**;**

**}**

bool Triangle**::**judge**()**

**{**

//abcacb bac bca cab cba

**if(**\_side1**+**\_side2**>**\_side3 **&&** \_side1**+**\_side3**>**\_side2 **&&** \_side2**+**\_side1**>**\_side3 **&&** \_side2**+**\_side3**>**\_side1 **&&** \_side3**+**\_side1**>**\_side2 **&&**\_side3**+**\_side2**>**\_side1**)**

**return true;**

**return false;**

**}**

//test.cpp

#include "Circle.h"

#include "Triangle.h"

#include "Rectangle.h"

/\*

为什么不能在main.cpp或者其他的的头文件中添加<windows.h>,否则第28到第31行（Rectangle 相关的）会报错？

\*/

int main**()**

**{**

Circle c1**(**1.0**);**

c1**.**show**();**

std**::**cout**<<**std**::**endl**;**

Triangle t1**(**1.0**,**2.0**,**3.0**);**

t1**.**show**();**

std**::**cout**<<**std**::**endl**;**

Triangle t2**(**10.0**,**3.0**,**8.0**);**

t2**.**show**();**

std**::**cout**<<**std**::**endl**;**

std**::**cout**<<**std**::**endl**;**

Triangle t3**(**3.0**,**4.0**,**5.0**);**

t3**.**show**();**

std**::**cout**<<**std**::**endl**;**

Rectangle r1**(**1.0**,**2.0**),**r2**(**3.0**,**5.0**);**

r1**.**show**();**

std**::**cout**<<**std**::**endl**;**

r2**.**show**();**

std**::**cout**<<**std**::**endl**;**

//std::cout<<"The screen will be closed in 60s."<<std::endl;

//Sleep(1000\*60);

**return** 0**;**

**}**

//题目（3）源代码：

//writing\_instruments.h

#ifndef WRITING\_INSTRUMENTS\_H\_INCLUDED

#define WRITING\_INSTRUMENTS\_H\_INCLUDED

#include <iostream>

#include <string>

#include <windows.h>

class writing\_instruments

**{**

protected**:**

double price**;**

bool used\_in\_exam**;**

bool me\_owned\_**;**

double service\_span**;**

std**::**string place\_of\_origin**;**

int num**;**

public**:**

writing\_instruments**();**

writing\_instruments**(**int**);**

virtual **~**writing\_instruments**();**

virtual void show**(){**std**::**cout**<<**"..."**<<**std**::**endl**;}**

virtual voidadd**(**int**);**

virtual void \_minus**(**int**);**

**};**

#endif // WRITING\_INSTRUMENTS\_H\_INCLUDED

//writing\_instruments.cpp

#include "writing\_instruments.h"

writing\_instruments**::**writing\_instruments**()**

**{**

price**=**0.0**;**

used\_in\_exam**=false;**

me\_owned\_**=false;**

service\_span**=**0.0**;**

place\_of\_origin**=**"xxxxxxxxxxxxxxxxx"**;**

num**=**0**;**

**}**

writing\_instruments**::~**writing\_instruments**()**

**{**

std**::**cout**<<**"The writing\_instruments destructor has been called."**<<**std**::**endl**;**

**}**

writing\_instruments**::**writing\_instruments**(**int n**)**

**{**

price**=**0.0**;**

used\_in\_exam**=false;**

me\_owned\_**=false;**

service\_span**=**0.0**;**

place\_of\_origin**=**"xxxxxxxxxxxxxxxxx"**;**

num**=**n**;**

**}**

void writing\_instruments**::**add**(**int n**)**

**{**

num**+=**n**;**

**}**

void writing\_instruments**::**\_minus**(**int n**)**

**{**

num**-=**n**;**

**}**

//pen.h

#ifndef PEN\_H\_INCLUDED

#define PEN\_H\_INCLUDED

#include "writing\_instruments.h"

class pen**:**publi cwriting\_instruments

**{**

private**:**

std**::**string nib**;**

std**::**string colour**;**

//enumcolour{Red,Orange,Yellow,Green,Blue,Indigo,Violet};

std**::**string kind**;**

public**:**

pen**(**int**,**std**::**string**,**std**::**string**);**

virtual **~**pen**();**

void show**();**

void add**(**int**);**

void \_minus**(**int**);**

**};**

#endif // PEN\_H\_INCLUDED

//pen.cpp

#include "pen.h"

pen**::**pen**(**int n**,**std**::**string \_colour**,**std**::**string \_nib**)**

**{**

price**=**10**;**

nib**=**\_nib**;**

num**=**n**;**

colour**=**\_colour**;**

price**=**100.0**;**

used\_in\_exam**=true;**

me\_owned\_**=true;**

service\_span**=**10.0**;**

place\_of\_origin**=**"China"**;**

**}**

pen**::~**pen**()**

**{**

std**::**cout**<<**"The pen destructor has been called."**<<**std**::**endl**;**

**}**

void pen**::**add**(**int n**)**

**{**

num**+=**10**\***n**;**

**}**

void pen**::**\_minus**(**int n**)**

**{**

num**-=**100**\*(-**n**);**

**}**

void pen**::**show**()**

**{**

add**(**10**);**

\_minus**(**20**);**

std**::**cout**<<**"It is a pen object."**<<**std**::**endl**;**

std**::**cout**<<**"Here are "**<<**num**<<**" pen "**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The nib of the pen is "**<<**nib**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The colour of the pen is "**<<**colour**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The kind of the pen is "**<<**kind**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The price of the pen is "**<<**price**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The pen can serve you for "**<<**service\_span**<<**" years."**<<**std**::**endl**;**

std**::**cout**<<**"The pen was made in "**<<**place\_of\_origin**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"You can use it in examination"**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"Some pens owned."**<<**std**::**endl**;**

**}**

//pencil.h

#ifndef PENCIL\_H\_INCLUDED

#define PENCIL\_H\_INCLUDED

#include "writing\_instruments.h"

class pencil**:**public writing\_instruments

**{**

private**:**

std**::**string colour**;**

std**::**string apply\_to**;**

std**::**string XB**;**

public**:**

pencil**(**int**,**std**::**string**,**std**::**string**);**

virtual **~**pencil**();**

void show**();**

void add**(**int**);**

void \_minus**(**int**);**

**};**

#endif // PENCIL\_H\_INCLUDED

//pencil.cpp

#include "pencil.h"

pencil**::**pencil**(**int n**,**std**::**string \_colour**,**std**::**string \_XB**)**

**{**

price**=**10**;**

num**=**n**;**

colour**=**\_colour**;**

XB**=**\_XB**;**

price**=**1.0**;**

used\_in\_exam**=true;**

me\_owned\_**=true;**

service\_span**=**0.3**;**

place\_of\_origin**=**"China"**;**

apply\_to**=**"students&&workmen&&white\_collar&&etc"**;**

**}**

pencil**::~**pencil**()**

**{**

std**::**cout**<<**"The pencil destructor has been called."**<<**std**::**endl**;**

**}**

void pencil**::**add**(**int n**)**

**{**

num**+=**120**\***n**;**

**}**

void pencil**::**\_minus**(**int n**)**

**{**

num**-=**100**\*(-**2**\***n**);**

**}**

void pencil**::**show**()**

**{**

add**(**10**);**

\_minus**(**20**);**

std**::**cout**<<**"It is a pencil object."**<<**std**::**endl**;**

std**::**cout**<<**"Here are "**<<**num**<<**" pencils "**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The colour of the pencil is "**<<**colour**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The price of the pencil is "**<<**price**<<**" yuan."**<<**std**::**endl**;**

std**::**cout**<<**"The pencil can serve you for "**<<**service\_span**<<**" years."**<<**std**::**endl**;**

std**::**cout**<<**"The pencil was made in "**<<**place\_of\_origin**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"You can use it in examination"**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"Some pencils owned."**<<**std**::**endl**;**

**}**

//sign\_pen.h

#ifndef SIGN\_PEN\_H\_INCLUDED

#define SIGN\_PEN\_H\_INCLUDED

#include "writing\_instruments.h"

class sign\_pen**:**public writing\_instruments

**{**

private**:**

std**::**string colour**;**

std**::**string apply\_to**;**

double nib\_diameter**;**

public**:**

sign\_pen**(**int**,**double**,**std**::**string**);**

virtual **~**sign\_pen**();**

void show**();**

void add**(**int**);**

void \_minus**(**int**);**

**};**

#endif // SIGN\_PEN\_H\_INCLUDED

//sign\_pen.cpp

#include "sign\_pen.h"

sign\_pen**::**sign\_pen**(**int n**,**double\_nib\_diameter**,**std**::**string \_colour**)**

**{**

nib\_diameter**=**\_nib\_diameter**;**

price**=**10**;**

num**=**n**;**

colour**=**\_colour**;**

price**=**1.0**;**

used\_in\_exam**=true;**

me\_owned\_**=true;**

service\_span**=**2.0**;**

place\_of\_origin**=**"China"**;**

apply\_to**=**"students&&workmen&&white\_collar&&etc"**;**

**}**

sign\_pen**::~**sign\_pen**()**

**{**

std**::**cout**<<**"The sign\_pen destructor has been called."**<<**std**::**endl**;**

**}**

void sign\_pen**::**add**(**int n**)**

**{**

num**+=**n**\***n**+**2**;**

**}**

void sign\_pen**::**\_minus**(**int n**)**

**{**

num**-=(**n**\***n**)/**2**;**

**}**

void sign\_pen**::**show**()**

**{**

add**(**109**);**

\_minus**(**20**);**

std**::**cout**<<**"It is a sign\_pen object."**<<**std**::**endl**;**

std**::**cout**<<**"Here are "**<<**num**<<**" sign pens "**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The colour of the sign pen is "**<<**colour**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The price of the sign pen is "**<<**price**<<**" yuan."**<<**std**::**endl**;**

std**::**cout**<<**"The diameter of the sign pen nib "**<<**nib\_diameter**<<**" mm."**<<**std**::**endl**;**

std**::**cout**<<**"The sign pen can serve you for "**<<**service\_span**<<**" years."**<<**std**::**endl**;**

std**::**cout**<<**"The sign pen was made in "**<<**place\_of\_origin**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"You can use it in examination"**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"Some sign pens owned."**<<**std**::**endl**;**

**}**

//writing\_brushes.h

#ifndef WRITING\_BRUSHES\_H\_INCLUDED

#define WRITING\_BRUSHES\_H\_INCLUDED

#include "writing\_instruments.h"

#include <cmath>

class writing\_brushes**:**public writing\_instruments

**{**

private**:**

std**::**string property**;**

std**::**string person\_apply\_to**;**

std**::**string apply\_to\_what**;**

std**::**string colour**;**

public**:**

writing\_brushes**(**int**,**std**::**string**,**std**::**string**,**std**::**string**);**

virtual **~**writing\_brushes**();**

void show**();**

void add**(**int**);**

void \_minus**(**int**);**

**};**

#endif // WRITING\_BRUSHES\_H\_INCLUDED

//writing\_brushes.cpp

#include "writing\_brushes.h"

writing\_brushes**::**writing\_brushes**(**int n**,**std**::**string \_colour**,**std**::**string \_property**,**std**::**string \_apply\_to\_what**)**

**{**

price**=**10**;**

property**=**\_property**;**

num**=**n**;**

colour**=**\_colour**;**

price**=**255.0**;**

used\_in\_exam**=false;**

me\_owned\_**=false;**

service\_span**=**2.0**;**

place\_of\_origin**=**"China"**;**

person\_apply\_to**=**"students&&workmen&&white\_collar&&etc"**;**

apply\_to\_what**=**\_apply\_to\_what**;**

**}**

writing\_brushes**::~**writing\_brushes**()**

**{**

std**::**cout**<<**"The writing\_brushes destructor has been called."**<<**std**::**endl**;**

**}**

void writing\_brushes**::**add**(**int n**)**

**{**

num**+=**sqrt**(**n**)\***20**-**1**;**

**}**

void writing\_brushes**::**\_minus**(**int n**)**

**{**

num**-=**n**\***sqrt**(**n**)+**2**;**

**}**

void writing\_brushes**::**show**()**

**{**

add**(**109**);**

\_minus**(**19**);**

std**::**cout**<<**"It is a writing\_brushes object."**<<**std**::**endl**;**

std**::**cout**<<**"Here are "**<<**num**<<**" writing brushes "**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The colour of the writing brush is "**<<**colour**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The price of the writing brush is "**<<**price**<<**" yuan."**<<**std**::**endl**;**

std**::**cout**<<**"This kind of brush is suitable for "**<<**apply\_to\_what**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"This property of writing brush is "**<<**property**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"The writing brush can serve you for "**<<**service\_span**<<**" years."**<<**std**::**endl**;**

std**::**cout**<<**"The writing brush was made in "**<<**place\_of\_origin**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"Generally You can't use it in examination"**<<**"."**<<**std**::**endl**;**

std**::**cout**<<**"No writing brush owned."**<<**std**::**endl**;**

**}**

//test.cpp

#include "pen.h"

#include "pencil.h"

#include "sign\_pen.h"

#include "writing\_brushes.h"

int main**()**

**{**

pen p1**(**10**,**"Black"**,**"Steel"**);**

p1**.**show**();**

std**::**cout**<<**std**::**endl**;**

pencil pc1**(**100**,**"Red"**,**"2B"**);**

pc1**.**show**();**

std**::**cout**<<**std**::**endl**;**

sign\_pen sgp1**(**20**,**0.5**,**"Purple"**);**

sgp1**.**show**();**

std**::**cout**<<**std**::**endl**;**

writing\_brushes wb1**(**25**,**"Black"**,**"hard"**,**"drawing"**);**

wb1**.**show**();**

std**::**cout**<<**std**::**endl**;**

std**::**cout**<<**"The screen will be closed in 60s."**<<**std**::**endl**;**

Sleep**(**1000**\***60**);**

**return**0**;**

**}**

//题目（4）源代码：

//\_complex\_number.h

#ifndef \_COMPLEX\_NUMBER\_H\_INCLUDED

#define \_COMPLEX\_NUMBER\_H\_INCLUDED

#include <iostream>

#include <cmath>

class \_complex\_number

**{**

private**:**

double re**;**

double im**;**

public**:**

\_complex\_number**();**

\_complex\_number**(**double**,**double**);**

**~**\_complex\_number**();**

void display**(**\_complex\_number**&);**

\_complex\_number **operator+(**\_complex\_number**&)**const**;**

\_complex\_number **operator-(**\_complex\_number**&)**const**;**

//\_complex\_number operator=(\_complex\_number&);//为什么这里用引用做参数就不行呢？

\_complex\_number **operator=(**\_complex\_number**);**

**};**

#endif // \_COMPLEX\_NUMBER\_H\_INCLUDED

//\_complex\_number.cpp

#include "\_complex\_number.h"

\_complex\_number**::**\_complex\_number**()**

**{**

**}**

\_complex\_number**::**\_complex\_number**(**double \_re**,**double \_im**)**

**{**

re**=**\_re**;**

im**=**\_im**;**

**}**

\_complex\_number**::~**\_complex\_number**()**

**{**

std**::**cout**<<**"The destructor has been called."**<<**std**::**endl**;**

**}**

/\*

注意，在运算符表示法中，运算符左侧的对象是调用对象，运算符右边的对象是作为参数被传递的对象。

\*/

\_complex\_number \_complex\_number**::operator+(**\_complex\_number**&**c1**)**const

**{**

\_complex\_number c2**;**

c2**.**re**=**re**+**c1**.**re**;**

c2**.**im**=**im**+**c1**.**im**;**

**return** c2**;**

**}**

\_complex\_number \_complex\_number**::operator-(**\_complex\_number**&**c1**)**const

**{**

\_complex\_number c2**;**

c2**.**re**=**re**-**c1**.**re**;**

c2**.**im**=**im**-**c1**.**im**;**

**return** c2**;**

**}**

\_complex\_number \_complex\_number**::operator=(**\_complex\_number c1**)**//有时需要考虑自赋值的特殊情况

**{**

//\_complex\_number c2;

re**=**c1**.**re**;**

im**=**c1**.**im**;**

**return\*this;**

**}**

void \_complex\_number**::**display**(**\_complex\_number**&**c**)**

**{**

**if(**c**.**im**>**0**)**//为什么这里写成 if(c.im>0) 输出与预期不符？

std**::**cout**<<**c**.**re**<<**" + "**<<**c**.**im**<<**"i"**<<**std**::**endl**;**

**else**

std**::**cout**<<**c**.**re**<<**" - "**<<**std**::**abs**(**c**.**im**)<<**"i"**<<**std**::**endl**;**

**}**

//test.cpp

#include "\_complex\_number.h"

int main**()**

**{**

\_complex\_number c1**(**1.0**,**2.0**),**c2**(**3.0**,**5.0**),**c3**,**c4**;**

c1**.**display**(**c1**);**

c2**.**display**(**c2**);**

c3**=**c1**-**c2**;**

c4**=**c1**+**c2**;**

c3**.**display**(**c3**);**

c4**.**display**(**c4**);**

Sleep**(**1000**\***60**);**

std**::**cout**<<**"The screen will be closed in 60s."**<<**std**::**endl**;**

**return** 0**;**

**}**