## 实验十

**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;};

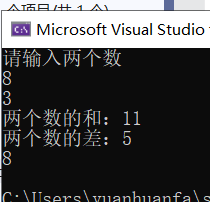
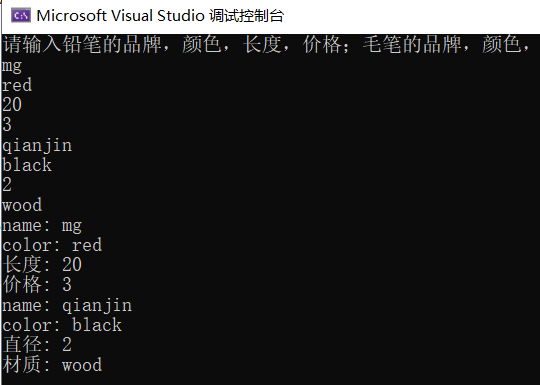
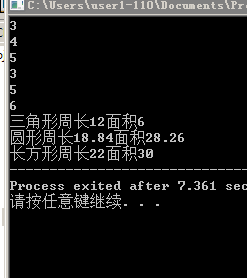
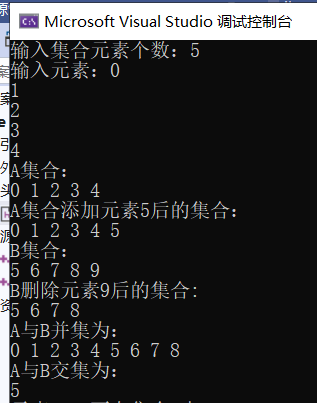
private:

}；

**（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**



**4 源代码**

**第一题**

#pragma once

#include <iostream>

using namespace *std*;

class jh

{

private:

int num[100];

int len;

public:

jh() { len = 0; };

jh(int data[], int n);

void get();

void select();

void show();

friend bool add(jh& A, int n);

friend void del(jh& A, int n);

friend jh jiao(jh& A,jh& B);

friend jh bing(jh& A, jh& B);

friend bool judge(jh& A, int n);

};

#include "jh.h"

jh::jh(int data[], int n)

{

len = n;

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

num[i] = data[i];

select();

}

void jh::get()

{

*cout* << "输入集合元素个数：";

*cin* >> len;

*cout* << "输入元素：";

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

*cin* >> num[i];

select();

}

void jh::select()

{

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

{

for (int j = i + 1; j < len; j++)

if (num[i] == num[j])

{

for (int k = j; k < len - 1; k++)

num[k] = num[k + 1];

len--;

j--;

}

}

}

void jh::show()

{

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

*cout* << num[i] << " ";

*cout* << *endl*;

}

bool add(jh& A, int n)

{

if (A.len == 100)

return false;

else

A.num[A.len++] = n;

A.select();

return true;

}

void del(jh& A, int n)

{

for (int i = 0; i < A.len; i++)

{

if (A.num[i] == n)

{

for (int j = i; j < A.len - 1; j++)

A.num[j] = A.num[j + 1];

A.len--;

break;

}

}

}

jh bing(jh& A, jh& B)

{

jh C;

C.len = A.len;

for (int i = 0; i < A.len; i++)

{

C.num[i] = A.num[i];

}

for (int i = 0; i < B.len; i++)

{

int index = 0;

for (index; index < A.len; index++)

{

if (A.num[index] == B.num[i])

break;

}

if (index == A.len)

C.num[C.len++] = B.num[i];

}

return C;

}

jh jiao(jh& A, jh& B)

{

jh C;

for (int i = 0; i < A.len; i++)

{

for (int j = 0; j < B.len; j++)

{

if (A.num[i] == B.num[j])

{

C.num[C.len++] = A.num[i];

break;

}

}

}

return C;

}

bool judge(jh& A, int n)

{

for (int i = 0; i < A.len; i++)

{

if (A.num[i] == n)

return true;

}

return false;

}

#include <iostream>

#include "jh.h"

int main()

{

int n[5] = {5,6,7,8,9 };

jh A, B(n, 5), C;

A.get();

*cout* << "A集合："<<*endl*;

A.show();

*cout* << "A集合添加元素5后的集合："<<*endl*;

add(A, 5);

A.show();

*cout* << "B集合：" << *endl*;

B.show();

*cout* << "B删除元素9后的集合: "<<*endl*;

del(B, 9);

B.show();

*cout* << "A与B并集为："<<*endl*;

C = bing(A, B);

C.show();

*cout* << "A与B交集为："<<*endl*;

C = jiao(A, B);

C.show();

if (judge(A, 2019))

*cout* << "元素2019在集合A中" << *endl*;

else

*cout* << "元素2019不在集合A中" << *endl*;

return 0;

}

### 第二题

#include "Shape.h"

#include<iostream>

using namespace std;

double Shape::area()

{return 0;}

double Shape::girth()

{return 0;}

void Shape::show()

{

cout<<"Shape Object"<<endl;

}

#ifndef SHAPE\_H

#define SHAPE\_H

class Shape

{

public:

double area();

double girth();

void show();

private:

};

#endif

#ifndef TRIANGLE\_H

#define TRIANGLE\_H

#include "Shape.h"

class Triangle : public Shape

{

public:

Triangle();

~Triangle();

double area(double ,double,double);

double girth(double ,double,double);

void show();

private:

double a,b,c,g,s;

};

#endif

#include "Triangle.h"

#include<iostream>

#include"math.h"

using namespace std;

Triangle::Triangle()

{

}

Triangle::~Triangle()

{

}

double Triangle::area(double a,double b,double c)

{

double p=(a+b+c)/2;

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

return s;

}

double Triangle::girth(double a,double b,double c)

{

g=a+b+c;

return g;

}

void Triangle::show()

{

cout<<"三角形周长"<<g<<"面积"<<s<<endl;

}

#define CIRCLE\_H

#include "Shape.h"

class Circle :

public Shape

{

public:

Circle();

~Circle(void);

double area(double );

double girth(double );

void show();

private:

double cs,r,cl;

};

#include "Circle.h"

#include<iostream>

#include"math.h"

using namespace std;

Circle::Circle()

{

}

Circle::~Circle(void)

{

}

double Circle::area(double r)

{

cs=3.14\*r\*r ;

return cs;

}

double Circle::girth(double r)

{

cl=2\*r\*3.14;

return cl;

}

void Circle::show()

{

cout<<"圆形周长"<<cl<<"面积"<<cs<<endl;

}

#ifndef RECTANGLE\_H

#define RECTANGLE\_H

#include "shape.h"

class Rectangle : public Shape

{

public:

Rectangle();

~Rectangle();

double area(double ,double);

double girth(double ,double);

void show();

private:

double x,y,rs,rl;

};

#endif

#include "Rectangle.h"

#include<iostream>

using namespace std;

Rectangle::Rectangle()

{

}

Rectangle::~Rectangle()

{

}

double Rectangle::area(double x,double y)

{

rs=2\*x\*y;

return rs;

}

double Rectangle::girth(double x,double y)

{ rl=x+y;

return rl;

}

void Rectangle::show()

{

cout<<"长方形周长"<<rl<<"面积" <<rs;

}

#include"Circle.h"

#include"Rectangle.h"

#include"Triangle.h"

#include <iostream>

using namespace std;

int main()

{

double a ,b ,c,r,x,y;

cin>>a>>b>>c>>r>>x>>y;

Triangle T1;

Circle C1;

Rectangle R1;

T1.area(a,b,c);

T1.girth(a,b,c);

T1.show();

C1.area(r);

C1.girth(r);

C1.show();

R1.area(x,y);

R1.girth(x,y);

R1.show();

return 0；

}

### 第三题

#pragma once

class pen

{

public:

pen(char\*n,char\*c);

~pen();

void show();

private:

char\* name;

char\* color;

};

#include "pen.h"

#include <iostream>

#include "string"

#pragma warning(disable:4996)

using namespace *std*;

pen::pen(char\*n,char\*c)

{

name = new char[100];

color = new char[100];

*strcpy*(name, n);

*strcpy*(color, c);

}

pen::~pen()

{

if (name!=*NULL*)

{

delete[]name;

}

if (color!=*NULL*)

{

delete[]color;

}

}

void pen::show()

{

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

*cout* << "color: " << color << *endl*;

}

#pragma once

#include"pen.h"

class pencil:public pen

{

public:

pencil(char\*n,char\*c,int\*l,int\* p);

~pencil();

void show();

private:

int\* length;

int\* price;

};

#include "pencil.h"

#include "string"

#include <iostream>

#pragma warning(disable:4996)

using namespace *std*;

pencil::pencil(char\* n, char\* c, int\* l, int\* p) :pen(n, c)

{

length = l;

price = p;

}

pencil::~pencil()

{

if (length!=*NULL*)

{

delete[]length;

}

if (price != *NULL*)

{

delete[]price;

}

}

void pencil::show()

{

pen::show();

*cout* << "长度: " <<\*length << *endl*;

*cout* << "价格: " << \*price << *endl*;

}

#pragma once

#include"pen.h"

class maopen:public pen

{

public:

maopen(char\* n, char\* c, int\* z, char\* cz);

~maopen();

void show();

private:

int\* zhijing;

char\*caizhi;

};

#include "maopen.h"

#include "string"

#include <iostream>

#pragma warning(disable:4996)

using namespace *std*;

maopen::maopen(char\* n, char\* c, int\* z, char\* cz) :pen(n, c)

{

zhijing=z;

caizhi = new char[20];

*strcpy*(caizhi, cz);

}

maopen::~maopen()

{

if (zhijing!= *NULL*)

{

delete[]zhijing;

}

if (caizhi!= *NULL*)

{

delete[]caizhi;

}

}

void maopen::show()

{

pen::show();

*cout* << "直径: " << \*zhijing << *endl*;

*cout* << "材质: " << caizhi<< *endl*;

}

#include <iostream>

#include "pencil.h"

#include "maopen.h"

using namespace *std*;

int main()

{

char \*name=new char[20];

char\* color=new char[10];

int\* length = new int[10];

int\* price = new int[20];

char\* name1 = new char[20];

char\* color1 = new char[10];

int\* zhijing = new int[10];

char\* caizhi = new char[20];

*cout* << "请输入铅笔的品牌，颜色，长度，价格；毛笔的品牌，颜色，直径，材质" << *endl*;

*cin* >> name >> color>> \*length>>\*price >> name1 >> color1 >>\*zhijing >> caizhi;

pencil s(name, color, length, price);

s.show();

maopen m(name1, color1, zhijing, caizhi);

m.show();

return 0;

}

### 第四题

#pragma once

class fushu

{

public:

fushu();

fushu(double x);

//fushu operator \* ( fushu& c2);

fushu operator ==(fushu& c2);

fushu operator + (fushu& c2);

fushu operator - (fushu& c2);

void display();

private:

double a;

};

#include "fushu.h"

#include <iostream>

using namespace *std*;

fushu::fushu()

{

}

fushu::fushu(double x)

{

a = x;

}

fushu fushu::operator+(fushu& c2)

{

return fushu(a + c2.a);

}

fushu fushu::operator-(fushu& c2)

{

return fushu(a - c2.a);

}

fushu fushu::operator==(fushu& c2)

{

return fushu(a = c2.a);

}

#include "fushu.h"

#include <iostream>

using namespace *std*;

int main()

{

double x, y;

*cout* << "请输入两个数" << *endl*;

*cin* >> x >> y;

fushu c1(x), c2(y), c3;

c3=c1 + c2;

*cout* << "两个数的和：";

c3.display();

*cout* << "两个数的差：";

c3=c1 - c2;

c3.display();

c3 == c1;

c3.display();

return 0;

}