**一程序代码**

#include<iostream>

#include<string>

using namespace std;

class MyArray {

public:

MyArray() {

cout << "MyArray类对象已创建!" << endl;

};

MyArray(int length);

~MyArray();

void Input();

void Display(string);

protected:

int \*alist;

int length;

};

MyArray::MyArray(int leng)

{

if (leng <= 0)

{

cout << "error length";

exit(1);

}

length = leng;

alist = new int[length];

if (alist == NULL)

{

cout << "assign failure";

exit(1);

}

cout << "MyArray类对象已创建!" << endl;

}

MyArray::~MyArray()

{

cout << "MyArray类对象已撤销!" << endl;

}

void MyArray::Display(string str)

{

int i;

int \*p=alist;

cout<<str<<length<<"个整数: "<<endl;

for(i=0;i<length;i++,p++)

cout<<\*p<<" ";

cout<<endl;

}

void MyArray::Input()

{

cout<<"请从键盘输入"<<length<<"个整数:";

int i;

int \*p=alist;

for(i=0;i<length;i++,p++)

cin>>\*p;

}

class SortArray:public MyArray

{

public:

SortArray(int leng);

~SortArray();

void Order();

};

SortArray::SortArray(int leng)

{

if (leng <= 0)

{

cout << "error length";

exit(1);

}

length = leng;

alist = new int[length];

if (alist == NULL)

{

cout << "assign failure";

exit(1);

}

cout << "SortArray类对象已创建!" << endl;

}

SortArray::~SortArray()

{

cout << "SortArray类对象已撤销!" << endl;

}

void SortArray::Order()

{

int i, t, j;

for (i = 0; i < length-1; i++)

{

for (j = 0; j < length-i-1; j++)

{

if (alist[j] > alist[j+1])

{

t = alist[j];

alist[j] = alist[j+1];

alist[j+1] = t;

}

}

}

}

int main()

{

SortArray a(5);

a.Input();

a.Display("显示已经输入的");

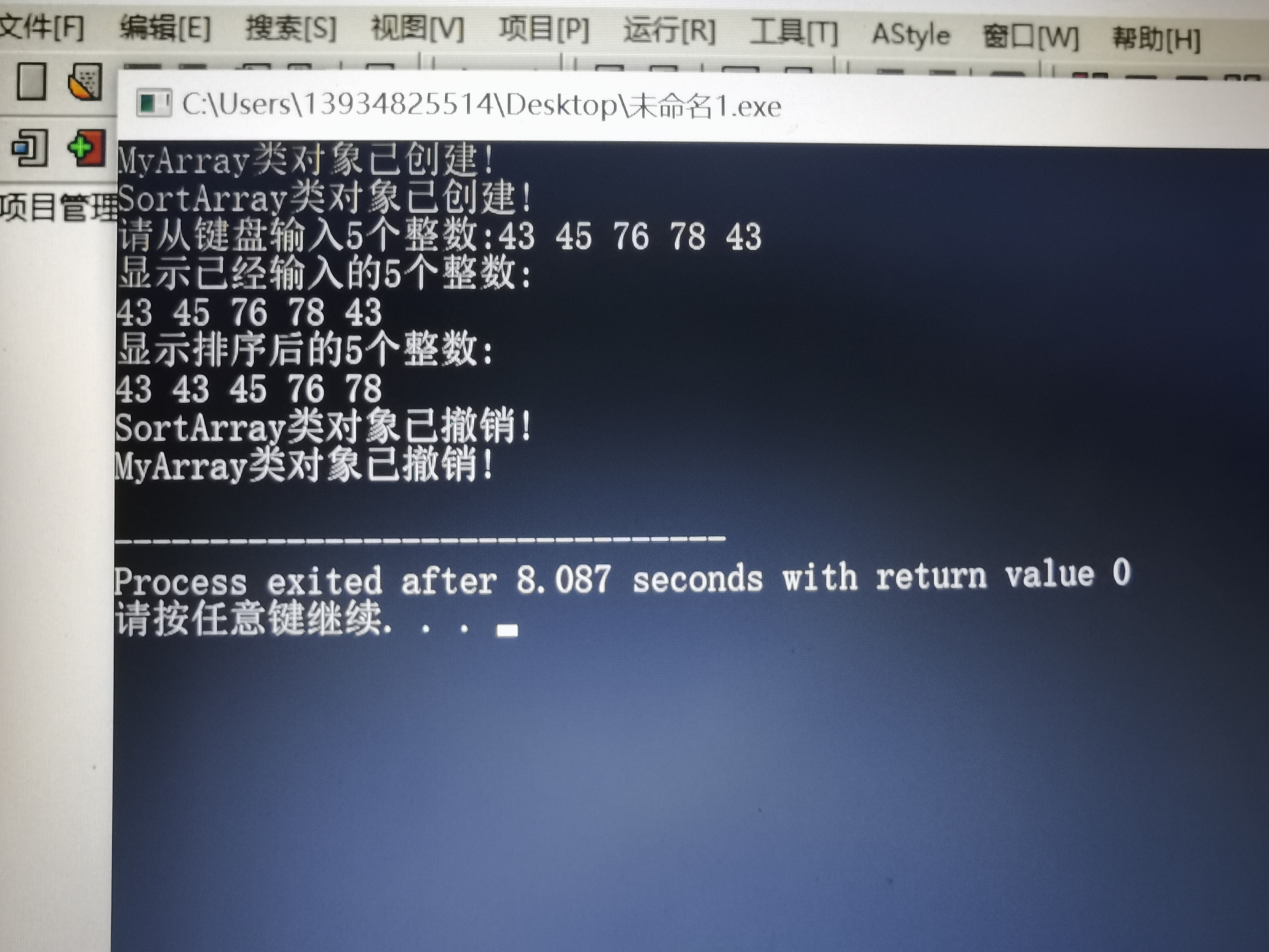
a.Order();

a.Display("显示排序后的");

return 0;

}

**二结果**



**三心得体会**

**1.程序在运行时，会先运行基类的构造函数，然后是派生类的构造函数，而析构函数正好相反，先是派生类的析构函数，再是基类的析构函数**

**2.派生类不会继承基类的构造函数和析构函数，需要在派生类中重新定义**

**3.在定义派生类对象时，构造函数执行顺序：**

**（1）调用基类的构造函数，对基类数据成员进行初始化**

**（2）调用内嵌对象成员函数的构造函数，对内嵌对象成员的数据成员初始化**

**（3）执行派生类的构造函数体，对派生类数据成员初始化**

Copyright ©2021-2099 lvlei. All rights reserved