上机实验（六）

#include<iostream>

#include<string>

using namespace std;

class MyArray {

public:

MyArray(int length); //202030310218李嘉杰

MyArray();

~MyArray();

void Input();

void Display(string);

int a[5];

protected:

int\* alist;

int length;

};

class SortArray :public MyArray

{

public:

SortArray(int a):MyArray(a)

{

length = a;

cout << "SortArray类对象已创建!" << endl;

}

~SortArray();

void dis(string);

void px();

private :

int b;

};

void SortArray::px()

{

int i, j, temp;

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

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

{

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

{

temp = a[j];

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

a[j + 1] = temp;

}

}

cout << "显示排序以后的的五个整数";

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

cout << a[i] << " ";

cout << endl;

}

void SortArray::dis(string str)

{

int i;

int\* p = alist;

cout << str << length << "个整数: ";

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

cout << \*p << " ";

cout << endl;

}

SortArray::~SortArray()

{

cout << "SortArray类对象已撤销!" << endl;

}

MyArray::MyArray()

{

}

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()

{

delete[] alist;

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: ";

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;

a[i] = \*p;

}

}

int main()

{

SortArray a(5);

a.Input();

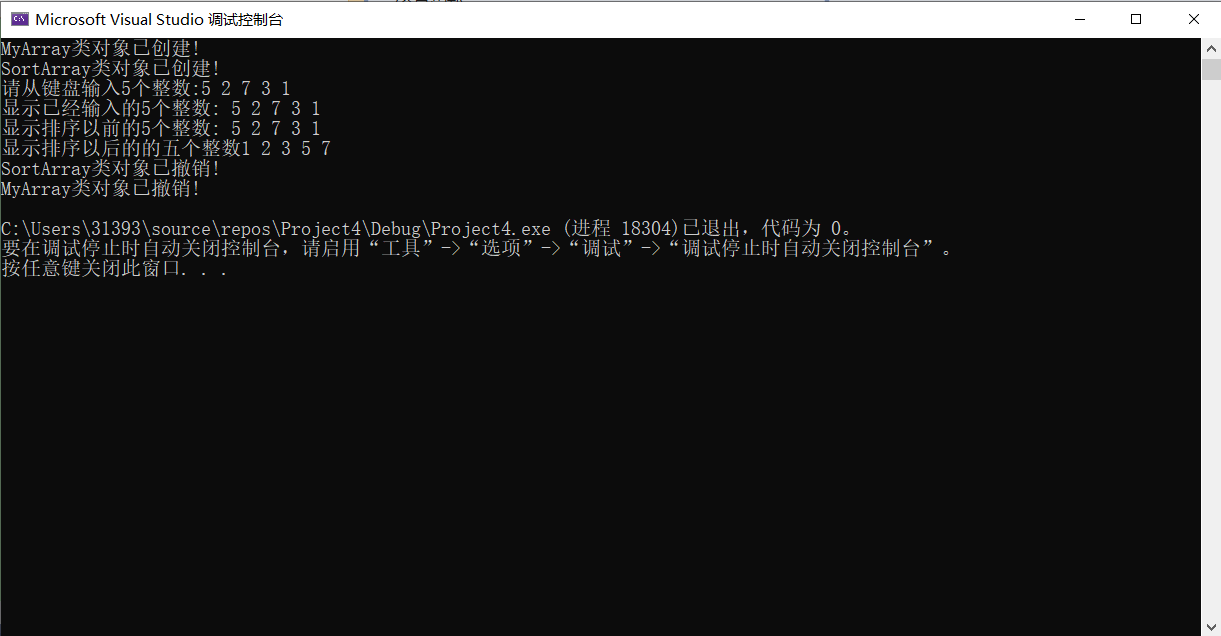
a.Display("显示已经输入的");

a.dis("显示排序以前的");

a.px();

return 0;

}



总结：

这次试验让我更好的理解构造函数与析构函数的执行顺序，从而使我们更好的理解派生类与基类的构造函数与析构函数的执行顺序。