Copyright 2021-2099 QinglinShan. All rights reserved

* **声明一个SortArray继承类MyArray，在该类中定义一个函数，具有将输入的整数从小到大进行排序的功能.**

**程序代码**

#include<iostream>

#include<string>

using namespace std;

class MyArray {

public:

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);

}

alist = new int[length];

length = leng;

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;

}

class SortArray : public MyArray {

public:

void Sort();

SortArray(int leng) :MyArray(leng)

{

cout << "SortArray类对象已创建!" << endl;

}

~SortArray();

};

SortArray::~SortArray()

{

cout << "SortArray类对象已撤销!" << endl;

}

void SortArray::Sort()

{

int i, j, temp;

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

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

{

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

{

temp = alist[j];

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

alist[j + 1] = temp;

}

}

}

int main()

{

SortArray s(5);

s.Input();

s.Display("显示排序以前的");

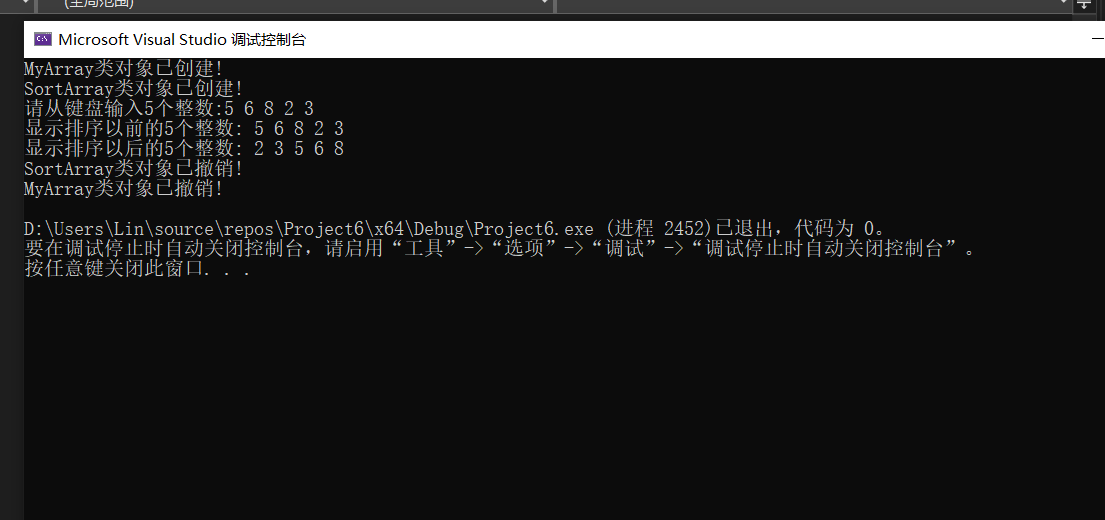
s.Sort();

s.Display("显示排序以后的");

return 0;

}

运行结果



感想心得

本次实验继承了上次实验的目的，更好的掌握了派生类的声明方法和派生类构造函数的定义方法。，还理解了构造函数和析构函数在基类和派生类中的执行顺序及构造规则：先调用基类的构造函数，再调用对象成员构造函数，最后调用派生类的构造函数。先调用派生类的构造函数，再调用对象成员构造函数，最后调用基类的构造函数；(与构造函数调用顺序相反)。同时我认为这也是所学内容中最难的部分之一，应该花费更多时间来学习这方面的知识。