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[leng];

length = leng;

if (alist == NULL)

{

cout << "assign failure";

exit(1);

}

cout << "MyArray类对象已创建。" << endl;

}

MyArray::~MyArray()

{

delete[] this->alist;

cout << "MyArray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 << "请从键盘中输入" << 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);

~SortArray();

void Sort();

};

void SortArray::Sort()

{

int i, j, temp;

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

{

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

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

{

temp = alist[j];

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

alist[j + 1] = temp;

}

}

}

SortArray::SortArray(int leng):MyArray(leng)

{

if (leng <= 0)

{

cout << "error length";

exit(1);

}

alist = new int[leng];

length = leng;

if (alist == NULL)

{

cout << "assign failure";

exit(1);

}

cout << "SortArray类对象已创建。" << endl;

}

SortArray::~SortArray()

{

/\*delete[] this->alist;\*/

cout << "SortArray对象已撤销！" << endl;

}

int main(void)

{

SortArray a(5);

a.input();

a.Display("显示排序以前的");

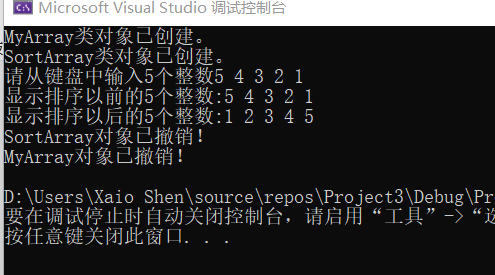
a.Sort();

a.Display("显示排序以后的");

return 0;

}

运行结果



感想心得

  通过对继承下的构造与析构函数的学习，我明白了：

1、子类对象在创建时需要调用父类构造函数进行初始化；

2、先执行父类构造函数然后执行成员的构造函数；

3、父类构造函数显示调用需要在初始化列表中进行；

4、子类对象在销毁时需要调用父类析构函数进行清理；

5、析构顺序与构造顺序对称相反。