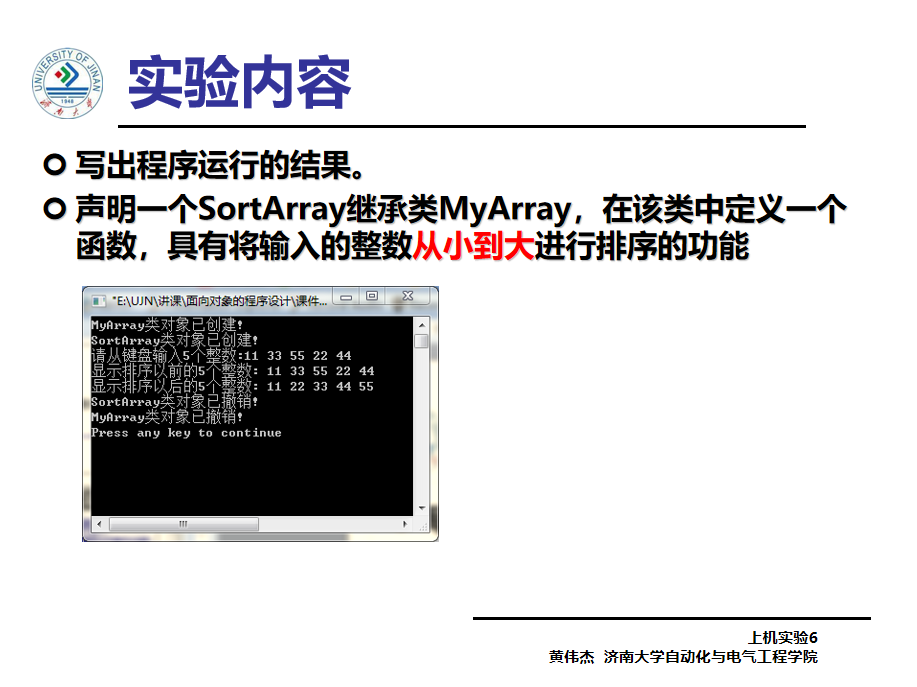
1. 实验题目

上机实验6：继承下的构造函数与析构函数



二、实验过程

————————————————代码：————————————————

#include<iostream>

#include<string>

using namespace std;

class MyArray {

public:

MyArray(int leng);

~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[] alist;

cout << "MyArray类对象已撤销!" << endl;

}

void MyArray::Display(string str)

{

int i;

cout << str << length << "个整数: ";

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

cout << \*(alist+i) << " ";

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;

}

//\*\*\*\*\*\*\*\*\*\*\*SortArray\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class SortArray :public MyArray {

public:

SortArray(int leng);

~SortArray();

void order();

private:

};

SortArray::SortArray(int leng)

:MyArray(leng) {

cout << "SortArray类对象已创建!" << endl;

}

SortArray::~SortArray() {

cout << "SortArray类对象已撤销!" << endl;

}

void SortArray::order() {

int k,t;

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

k = i;

for (int j = i + 1; j < length; j++) {

if (\*(alist + k) > \*(alist + j)) {

k = j;

}

}

if (k != i) {

t = \*(alist + k);

\*(alist + k) = \*(alist + i);

\*(alist + i) = t;

}

}

}

int main() {

SortArray a(5);

a.Input();

a.Display("显示排序以前的");

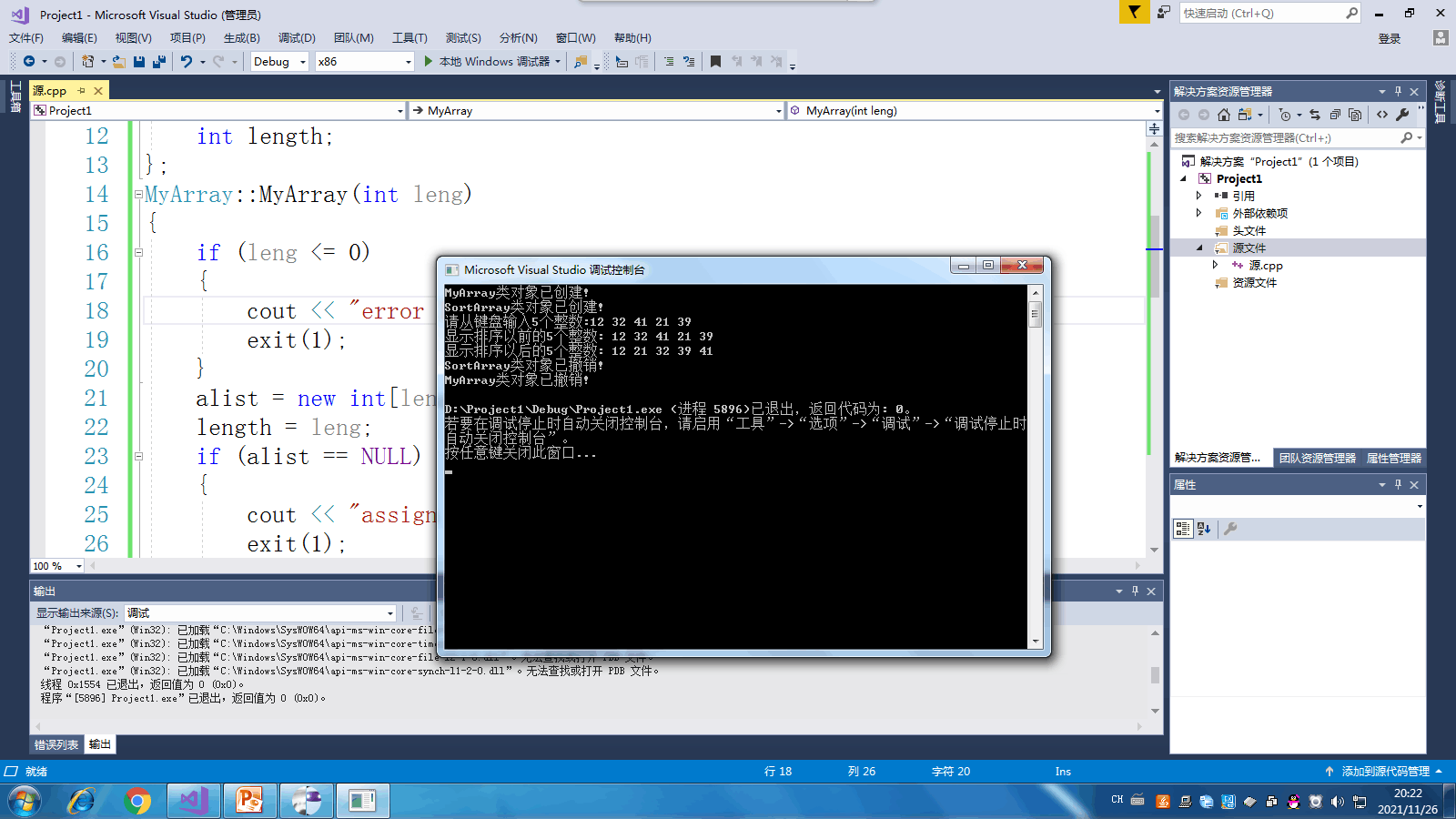
a.order();

a.Display("显示排序以后的");

return 0;

};

———————————————运行结果：———————————————



三、问题分析与感想

1、MyArray类中对输入的leng 进行了异常处理，保证输入的值不是小于等于零的，保证成功初始化数组。

2、第一次写的时候，主函数中没有正确定义变量，导致类中的数组长度越界，发生错误，改正后运行正常。

3、两次输出数据可以都使用基类里的display成员函数，只需在排序前后分别调用即可。

电自2002许吴瑕 202030310293

Copyright ©2021-2099 WuxiaXu. All right reserved