第六次上机实验

**1、程序代码**

**实验一**

#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);

}

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;

}

class SortArray :public MyArray

{

public:

int z;

SortArray(int leng):MyArray(leng)

{

cout << "SortArray类对象已创建!" << endl;

}

~SortArray()

{

cout << "SortArray类对象已撤销!" << endl;

}

void paixu()

{

int\* p = alist;

int x;

for (int i = length -1; i > 0; i--)

{

for (int j = 0; j < i; j++)

{

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

{

x = p[j];

p[j] = p[j + 1];

p[j + 1] = x;

}

}

}

}

};

int main()

{

SortArray a(5);

a.Input();

a.Display("显示已经输入的");

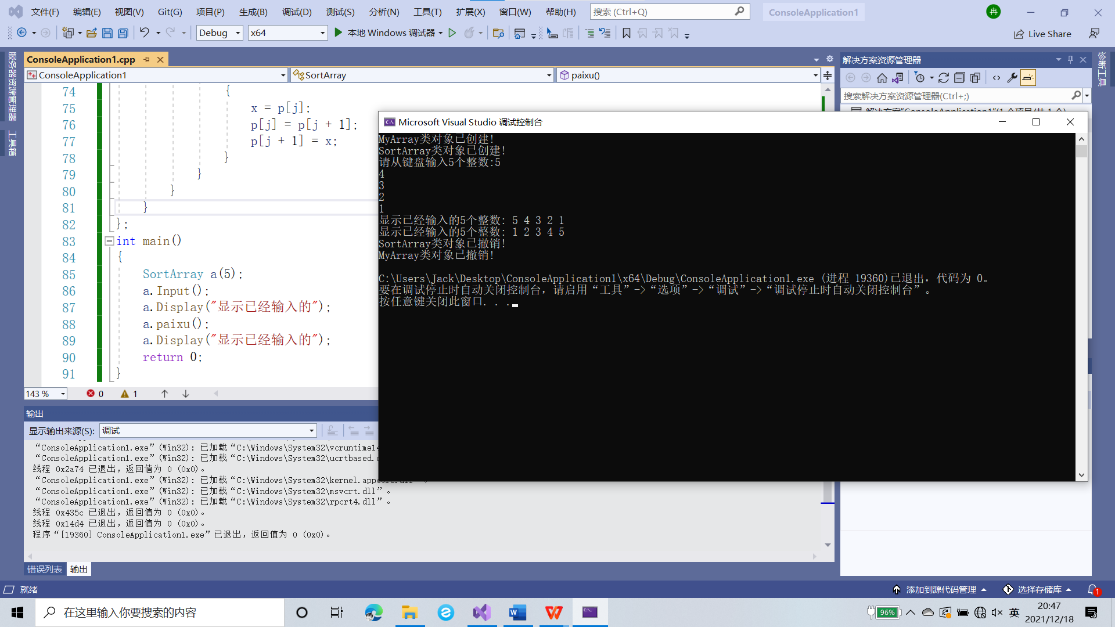
a.paixu();

a.Display("显示已经输入的");

return 0;

}

**2、程序结果**

**实验一**

**3、感想心得**

通过本次上机实验，我主要明白了派生类的构造函数与析构函数的构造规则，当基类构造函数没有参数或者没有显式定义时，派生类的构造函数无需传递参数，甚至无需定义。但是基类需要传递参数时，我们就应该按照格式定义派生类函数，然后在函数体内再增添自己新需要的语句。而析构函数则没有什么要求，按照正常的定义方式构造即可。

通过本次实验，我们学习到了C++三大基本特征之一——继承，继承可以省去许多时间，不用再重新定义一遍基类，直接采用现成的基类，在此基础上再增添新的内容即可，极大的减少了我们浪费的时间。