1. 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);

}

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:

void px();

SortArray(int leng);

~SortArray();

};

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::px()

{

int \*p = alist;

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

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

if (alist[i] < alist[j])

{

int t = p[i];

p[i] = p[j];

p[j] = t;

}

}

int main()

{

SortArray a(5);

a.Input();

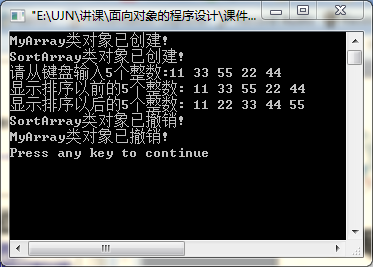
a.Display("显示排序以前的");

a.px();

a.Display("显示排序之后的");

return 0;

1. 实验结果



1. 实验感想

本次实验学习了派生类的声明方法和派生类构造函数的定义方法，编写了MyArray类的一个子类SortArray来实现对输入数据的排序。   继承是指一个子类(或称为派生类)继承父类(或称为基类)的特征(属性和操作)。继承是面向对象程序设计时实现代码复用的重要手段，它允许在原有的类的基础上进行扩展，增加功能，这样新产生的类称为子类。   派生类不能继承基类中的构造函数和析构函数。当基类含有带参数的构造函数时，派生类必须定义构造函数，以提供把参数传递给基类构造函数的途径。在C++的类继承中，构造函数不能被继承，建立对象时，首先调用基类的构造函数，然后在调用下一个派生类的构造函数，依次类推；调用析构对象的顺序正好与构造相反。