一、程序代码

#include<iostream>

#include<string>

using namespace std;

class MyArray {

public:

void Input();

void Display(string);

protected:

int\* alist;

int length;

};

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:

SortArray(int length);

~SortArray();

void sort();

};

SortArray::SortArray(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 << "SortArray类对象已创建!" << endl;

}

SortArray::~SortArray()

{

delete[] alist;

cout << "SortArray类对象已撤销!" << endl;

}

void SortArray::sort()

{

int temp;

int\* p = alist;

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

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

if (p[i] > p[j])

{

temp = p[i];

p[i] = p[j];

p[j] = temp;

}

cout << "排序后的：" ;

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

cout << p[i] << " ";

cout << endl;

}

int main()

{

SortArray a(5);

a.Input();

a.Display("显示已经输入的");

a.sort();

return 0;

}、

二、程序结果



三、感悟心得

1.在使用冒泡排序时，要注意循环结束的条件，防止出现循环次数太多或太少的问题。

2.构造函数和析构函数确实不能继承

3.new和delete一定要配合使用

4.有些报错看不懂百度上能提供很大帮助

Copyright ©2021-2099 ZhongkaiCui. All rights reserved