//继承中的构造和析构的顺序

//先构造父类再构造子类

//析构相反

#include<iostream>

using namespace std;

class base {

public:

base() {

cout << "base的构造" << endl;

}

~base(){

cout << "base的析构函数" << endl;

}

};

class son :public base {

public:

son() {

cout << "son的构造函数" << endl;

}

~son() {

cout << "son析构函数" << endl;

}

};

void test01() {

son b;

}

int main() {

test01();

system("pasue");

return 0;

}