实验11 综合实验(三)

一、实验目的

深入掌握面向对象编程方法。

二、实验任务

进一步完善实验10。在对象赋值或拷贝时，如果没有建立所有权概念，直接对指针成员赋值或拷贝（即浅赋值或浅拷贝），程序将试图删除指针所指对象两次，出现错误。为了解决该问题，可以两个指针将指向不同的对象，其中的一个对象是另一个对象的拷贝（即深赋值或深拷贝），缺点是浪费空间。为了提供效率，另外一种方案是，引入含引用计数概念，让不同的指针均指向同一块内存，同时记录指向该内存的指针个数，在最后指向该内存的指针个数为零时，释放该内存。重新实现share类的源代码。

1）给定智能指针类的定义和声明改为：

class share {

public:

share();

explicit share(A\* p1);

~share();

share(share& t2);

share& operator =(share& t2);

A\* operator ->();

A& operator \*();

private:

A\* point;

int\* count; //引入应用计数, 注意此处不能为int count; 思考一下原因

};

2)支持如下测试代码：

int main()

{

share sp(new A());

sp->n = 10;

cout << "sp->n is :" << sp->n << endl;

share sp2(sp);

sp2->n = 20;

cout << "sp->n is :" << sp2->n << endl;

share sp3;

sp3 = sp2;

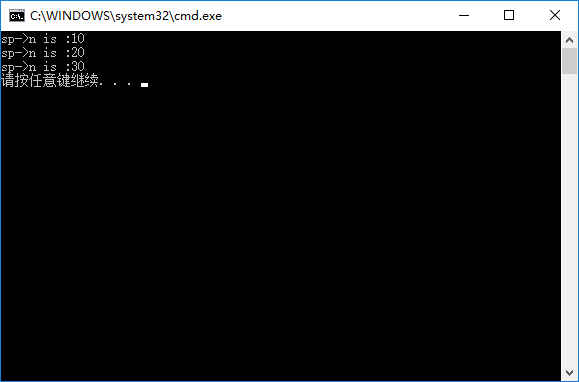
sp3->n = 30;

cout << "sp->n is :" << sp2->n << endl;

return 0;

}

输出结果为：



3）如果更换测试代码为下面的代码，又会如何？why?

int main() {

share sp(new A());

sp->n = 10;

cout << "sp->n is :" << sp->n << endl;

share sp2(sp);

sp2->n = 20;

cout << "sp2->n is :" << sp2->n << endl;

share sp3(new A());

sp3->n = 30;

sp = sp3;

cout << "sp->n is :" << sp->n << endl;

A a;

a = \*sp2;

cout << "a.n is :" << a.n << endl;

return 0;

}

三、实验要求

1. 对源文件的要求：有注释，有输入提示，分层缩进、隔行书写。
2. 解决方案的名称为：Lab11，每一小题为一个工程项目，取名为：Pro。
3. 删除解决方案文件夹中的所有Debug或.vs目录，然后打成压缩包（命名为“学号姓名.rar或zip”），提交至ftp://10.21.11.21上的“学生作业上传”里的相应目录。