

# 深拷贝与浅拷贝

2015年11月29日 星期日

下午4:21

## 一，简介

浅拷贝和深拷贝主要区别就是复制指针时是否重新创建内存空间。

## 二，详述

### 1，深拷贝浅拷贝（更偏向c++）

浅拷贝和深拷贝主要区别就是复制指针时是否重新创建内存空间。

在类与类直接拷贝的过程中，浅拷贝会将源对象中的成员变量直接赋值给目标对象中的对应的成员变量

一个类中如果，但如果这个类中有指针，只会把这个指针的值考过去，而指针对应的空间不会再拷贝一份新的。

深拷贝不同浅拷贝的地方是，将指针对应的空间再拷贝了一份。

有的文章中把浅拷贝叫做位拷贝，深拷贝叫做值拷贝。

### 2，c++

使用：

拷贝构造函数执行的是浅拷贝，如果要用深拷贝的话，可能要自己手写

### 3，python

#### 3.1 原理及实现：

由于python中，所有变量传参等都是传的引用，那么按照上面的对深拷贝浅拷贝的介绍，对类执行浅拷贝的之后，拷贝的源对象和目标对象中成员变量的地址都是一样的，改一个的话，另一个也跟着变。这个要注意！！

#### 3.2使用：

对类，直接用=赋值的话，是浅拷贝；如果要深拷贝，要import copy 调用a = copy.deepcopy (b)

#### 3.3实验：

实验路径：work@nj01-nlp-

test01.nj01.baidu.com:/home/work/tianzhiliang/test/cpp/copy

实验细节见附图

## 三，参考资料：

深拷贝浅拷贝介绍 不

错 <http://blog.csdn.net/xdx2ct1314/article/details/7495539>

深拷贝浅拷贝介绍 百度经验

simple <http://jingyan.baidu.com/article/2c8c281df2bed20008252ab8.html>

python拷贝 [http://blog.sina.com.cn/s/blog\\_81ad0841010172s9.html](http://blog.sina.com.cn/s/blog_81ad0841010172s9.html)

图1: python实验代码

```
1 import sys
2
3 class A:
4     #def __init__(self,a1,a2):
5     #     self.a1 = a1
6     #     self.a2 = a2
7     def __init__(self):
8         self.a1 = -1
9         self.a2 = -2
10
11 a=A()
12 b=A()
13 a.a1=1
14 print "a",a.a1,a.a2
15 b=a
16 print "a",a.a1,a.a2
17 print "b",b.a1,b.a2
18 a.a2=3
19 print "a",a.a1,a.a2
20 print "b",b.a1,b.a2
```

图2: python实验结果

```
[work@nj01-nlp-test01.nj01.baidu.com copy]$ python copy.py
a 1 -2
a 1 -2
b 1 -2
a 1 3
b 1 3
```