

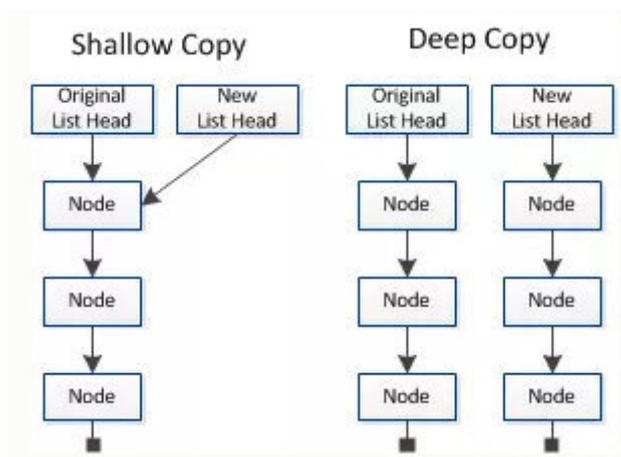
## 问题

深拷贝和浅拷贝的区别是面试中的常见问题之一，对于不同的编程语言，这个问题的回答可能稍有差别，下面我们就来探索一下它们之间的异同吧。

先来看看在JavaScript对象的深拷贝与浅拷贝的区别：

**浅拷贝**：只是复制了指向某个对象的指针，而不是复制对象本身，因此浅拷贝后新旧对象还是**共用同一块内存**，旧对象改变**会修改新对象的值**。

**深拷贝**：会另外构造一个与旧对象一模一样的对象，新旧对象**不共享内存**，旧对象改变**不会修改新对象的值**。



理解了上面的内容，可以说已经理解了深拷贝和浅拷贝的本质区别了。但是，如果面试官是问你在C++中深拷贝和浅拷贝是和拷贝的区别，上面的回答可能还不能让面试官满意哦。在C++中，要说出深拷贝和浅拷贝的区别，想必是需要提到拷贝构造函数这个概念才能让面试官满意的。既然如此，那我们就来看看C++中怎么说的吧。

## 浅拷贝

浅拷贝，也称为位拷贝。C++中的浅拷贝是通过拷贝构造函数来实现的，如果程序员不主动编写拷贝构造函数和赋值函数，编译器将以浅拷贝的方式自动生成缺省的函数，也就是在拷贝时简单地复制某个对象的指针，这样很容易造成一些问题。

例如，假设String类有两个对象a和b，a.data的内容为“hello”，b.data为“world”，当将a的值赋给b时，可能会出现以下3个问题：

- ① b.data的内存没释放，**造成内存泄漏**
- ② b.data和a.data指向了同一块内存，a或b**任何一方的值改变都会修改另一方的值**
- ③ **在对象被析构时，data被释放了两次**

看看下面的代码来理解一下：

```
1 //假设有一个String类
2 String a("hello");
3 String b("world");
4 //调用了拷贝构造函数，不过这里最好写成 string c(a)，a原本的内存没有释放，且a或c改变都会影响另一方
5 String c = a;
6 c=b; //调用了赋值函数
```

# 深拷贝

---

深拷贝**必须显示地提供拷贝构造函数和赋值运算符，而且新旧对象不共享内存**，也就是说，在编写拷贝构造函数时会开辟一个新的内存空间。那什么时候会使用深拷贝？

- ①一个对象以值传递的方式传入函数体
- ②一个对象以值传递的方式从函数体返回
- ③一个对象需要通过另外一个对象进行初始化

## 总结

---

C++中，浅拷贝不需要自己实现，编译器会自动生成缺省的拷贝构造函数，浅拷贝新旧对象共享一块内存，任何一方的值改变都会影响另一方；深拷贝需要自己手动编写拷贝构造函数，深拷贝新旧对象不共享内存。

## 参考资料

---

[浅拷贝与深拷贝的区别](#)

[C++细节 深拷贝和浅拷贝\(位拷贝\) 详解](#)