**深度解析Objective-C笔试题**

2011-08-11 17:39 佚名 互联网 [我要评论(0)](http://mobile.51cto.com/iphone-282872.htm#commment) 字号：[**T**](javascript:setfont(12);) | [**T**](javascript:setfont(16);)

[一键收藏，随时查看，分享好友！](javascript:favorBox('open');)

本文介绍的是Objective-C笔试题，先来问一个，为什么很多内置类如UITableViewController的delegate属性都是assign而不是retain的？看本文详细详细解答内容。

AD：

**Objective-C笔试题**是本文要介绍的内容，很详细的讲解写的答案。大约有18个**Objective-C**问题供你参考学习，不多说，我们一起来看详细解答！

1.Objective-C中，与alloc语义相反的方法是dealloc还是release？与retain语义相反的方法是dealloc还是release，为什么？需要与alloc配对使用的方法是dealloc还是release，为什么？

答：alloc与dealloc语意相反，alloc是创建变量，dealloc是释放变量。 retain 对应release,retain 保留一个对象。调用之后，变量的计数加1。或许不是很明显，在这有例为证：

1. - (void) setName : (NSString\*) name {
2. [name retain];
3. [myname release];
4. myname = name;
5. }

我们来解释一下：设想，用户在调用这个函数的时候，他注意了内存的管理，所以他小心的写了如下代码：

1. NSString \* newname = [[NSString alloc] initWithString: @"John"];
2. [aClass setName: newname];
3. [newname release];

我们来看一看newname的计数是怎么变化的。首先，它被alloc，count = 1; 然后，在setName中，它被retain， count = 2; 最后，用户自己释放newname，count = 1，myname指向了newname。这也解释了为什么需要调用[myname release]。我们需要在给myname赋新值的时候，释放掉以前老的变量。retain 之后直接dealloc对象计数器没有释放。alloc 需要与release配对使用，因为alloc 这个函数调用之后，变量的计数加1。所以在调用alloc 之后，一定要调用对应的release。另外，在release一个变量之后，他的值仍然有效，所以最好是后面紧接着再var = nil。

2.在一个对象的方法里面:

self.name = “object”;   
和   
name ＝”object”

有什么不同吗?

答：self.name = "object"会调用对象的setName()方法，name = "object"会直接把object赋值给当前对象的name 属性。

[backcolor=transparent][backcolor=transparent]

3.这段代码有什么问题吗:

1. [backcolor=transparent]@implementation Person
2. [backcolor=transparent]- (void)setAge:(int)newAge {
3. [backcolor=transparent]self.age = newAge;
4. [backcolor=transparent]}
5. [backcolor=transparent]
6. @end

答：会进入死循环。

4.什么是retain count?

答：引用计数(ref count或者retain count)。对象的内部保存一个数字，表示被引用的次数。例如，某个对象被两个指针所指向（引用）那么它的retain count为2。需要销毁对 象的时候，不直接调用dealloc，而是调用release。release会 让retain count减1，只有retain count等于0，系统才会调用dealloc真正销毁这个对象。

5.以下每行代码执行后，person对象的retain count分别是多少

1. Person \*person = [[Person alloc] init]; count 1
2. [person retain]; count 2
3. [person release];count 1
4. [person release];retain count = 1;

6.为什么很多内置类如UITableViewController的delegate属性都是assign而不是retain的？

答：会引起循环引用。

7.定义属性时，什么情况使用copy，assign，和retain 。

答：assign用于简单数据类型，如NSInteger,double,bool,retain 和copy用户对象，copy用于当 a指向一个对象，b也想指向同样的对象的时候，如果用assign，a如果释放，再调用b会crash,如果用copy 的方式，a和b各自有自己的内存，就可以解决这个问题。retain 会使计数器加一，也可以解决assign的问题。另外：atomic和nonatomic用来决定编译器生成的getter和setter是否为原子操作。在多线程环境下，原子操作是必要的，否则有可能引起错误的结果。加了atomic，setter函数会变成下面这样：

1. if (property != newValue) {
2. [property release];
3. property = [newValue retain];
4. }

8.的对象是在什么时候被release的？

答：autorelease实际上只是把对release的调用延迟了，对于每一个Autorelease，系统只是把该Object放入了当前的Autorelease pool中，当该pool被释放时，该pool中的所有Object会被调用Release。对于每一个Runloop，系统会隐式创建一个Autorelease pool，这样所有的release pool会构成一个象CallStack一样的一个栈式结构，在每一个Runloop结束时，当前栈顶的Autorelease pool会被销毁，这样这个pool里的每个Object（就是autorelease的对象）会被release。那什么是一个Runloop呢？一个UI事件，Timer call， delegate call， 都会是一个新的Runloop。那什么是一个Runloop呢？一个UI事件，Timer call， delegate call，都会是一个新的Runloop。

9.这段代码有什么问题,如何修改

1. for (int i = 0; i **<** **someLargeNumber**; i++)
2. {
3. NSString \*string = @”Abc”;
4. string = [string lowercaseString];
5. string = [string stringByAppendingString:@"xyz"];
6. NSLog(@“%@”, string);
7. }

答：会内存泄露，

1. for(int i = 0; i**<1000**;i++){
2. NSAutoreleasePool \* pool1 = [[NSAutoreleasePool alloc] init];
3. NSString \*string = @"Abc";
4. string = [string lowercaseString];
5. string = [string stringByAppendingString:@"xyz"];
6. NSLog(@"%@",string);
7. [pool1 drain];
8. }

10.autorelease和垃圾回收机制(gc)有什么关系？

答案：不懂

11.IPhone OS有没有垃圾回收（gc）？

没有

12.什么是Notification？

答：观察者模式，controller向defaultNotificationCenter添加自己的notification，其他类注册这个notification就可以收到通知，这些类可以在收到通知时做自己的操作（多观察者默认随机顺序发通知给观察者们，而且每个观察者都要等当前的某个观察者的操作做完才能轮到他来操作，可以用NotificationQueue的方式安排观察者的反应顺序，也可以在添加观察者中设定反映时间，取消观察需要在viewDidUnload 跟dealloc中都要注销）。参考链接：http://useyourloaf.com/blog/2010/6/6/delegation-or-notification.html

13.什么时候用delegate，什么时候用Notification？

答：delegate针对one-to-one关系，并且reciever可以返回值给sender，notification 可以针对one-to-one/many/none,reciever无法返回值给sender.所以，delegate用于sender希望接受到reciever的某个功能反馈值，notification用于通知多个object某个事件。

14.什么是KVC和KVO？

答：KVC(Key-Value-Coding)内部的实现：一个对象在调用setValue的时候，（1）首先根据方法名找到运行方法的时候所需要的环境参数。（2）他会从自己isa指针结合环境参数，找到具体的方法实现的接口。（3）再直接查找得来的具体的方法实现。KVO（Key-Value-Observing）：当观察者为一个对象的属性进行了注册，被观察对象的isa指针被修改的时候，isa指针就会指向一个中间类，而不是真实的类。所以isa指针其实不需要指向实例对象真实的类。所以我们的程序最好不要依赖于isa指针。在调用类的方法的时候，最好要明确对象实例的类名。

15.Notification和KVO有什么不同？

答：不知道

16.KVO在Objective-C中是怎么实现的？

答：不知道

17.ViewController 的 loadView, viewDidLoad, viewDidUnload 分别是在什么时候调用的？在自定义ViewController的时候这几个函数里面应该做什么工作？

答：viewDidLoad在view 从nib文件初始化时调用，loadView在controller的view为nil时调用。此方法在编程实现view时调用,view 控制器默认会注册memory warning notification,当view controller的任何view 没有用的时候，viewDidUnload会被调用，在这里实现将retain 的view release,如果是retain的IBOutlet view 属性则不要在这里release,IBOutlet会负责release 。

18.ViewController 的 didReceiveMemoryWarning 是在什么时候被调用的？默认的操作是什么?

答：默认调用[super didReceiveMemoryWarning]