朱祁林 zhuqilin

Cocoa And Dotnet.

我沉静我思考 我的新浪围脖
http://t.sina.com.cn/happy2016 欢迎加入

昵称：麒麟
园龄：6年9个月
荣誉：推荐博客
粉丝：933
关注：54
+加关注

< 2011年3月 >

日	一	二	三	四	五	六
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签
更多链接

最新随笔

1. 10分钟教你打造一个微信语音点歌系统
2. HTML5 移动浏览器支持

objective-c内存管理基础

对于我们.net开发人员来说，.net为我们提供了自动内存管理的机制，我们不需去关心内存的管理。但是iphone开发中却是不能的。这篇文章将简述一下objective-c的内存管理机制和方法和一些特性。

手动的进行内存管理

Cocoa和Objective-C的类都是NSObject的子类。NSObject中有几个方法进行内存管理。alloc方法为对象分配一片内存空间。dealloc方法用于释放对象的空间。但是在我们的代码中将永远都不会使用dealloc方法，因为运行时会为你调用此方法释放内存空间。而你需要做的只是引用计数，稍后介绍什么是引用计数。

除了alloc和dealloc，NSObject的还有retain和release方法两个方法用于引用计数。retain方法给retainCount变量加1，release方法给retainCount变量减1。当使用alloc为对象分配一片内存空间的时候，retainCount会为1。在这个对象的生命周期内，这个对象可能继续被其它变量引用。但有新的变量指向这个对象的时候，你应该调用retain方法，这样运行时才会知道有新的引用指向了这个变量，在这个对象生存期中拥有它的使用权。这个被Objective-C开发人员称之为"拥有"。例如：

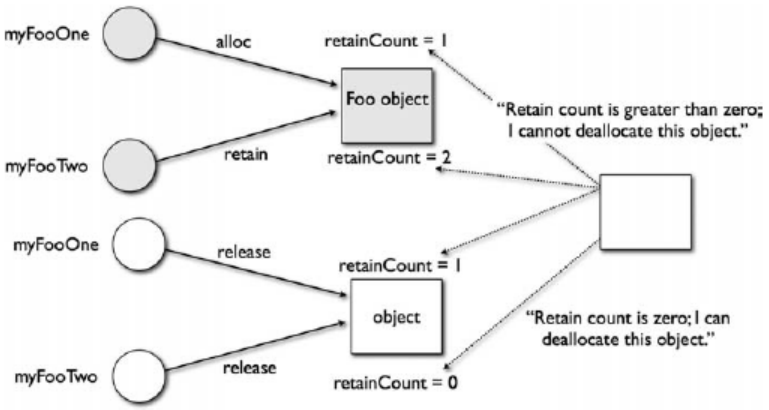
```
Foo * myFooOne = [[Foo alloc] init]; //retaincount 为1
Foo * myFooTwo = myFooOne; //myFooTwo 指向了这个对象
//retaincount 仍然为1
[myFooTwo retain]; //调用retain方法，运行时才知道myFooTwo指向了该对象,retaincount 为2
```

上面的代码中，myFooTwo通过调用retain方法，取得了Foo对象的拥有权。在这个对象的生命周期中，会有很多变量来指向和引用它。指向这个对象的变量也可以通过release方法来解除这种拥有权。release方法将会告诉运行时，我已经使用完这个变量了，已经不需要它了，retainCount计数减1。

当对象的retainCount的计数大于或者等于1的时候，运行时会继续维持这个对象。当对象的retainCount为0的时候，运行时会释放这个对象，并回收它占得内存空间。

下图展示了一个Foo对象的生命周期。Foo对象首先在内存中分配一个内存空间，并且被myFooOne引用。在这个时候Foo对象的retaincount为1。

```
Foo * myFooOne = [[Foo alloc] init];
```



第二个引用变量指向Foo对象，这个引用变量接着调用retain方法，其实也是调用Foo对象的retain方法。Foo对象的retaincount变成2。

```
Foo * myFooTwo = myFooOne;
[myFooTwo retain];
```

接着当myFooOne引用不需要的时候，通过调用release方法，解除与Foo对象的拥有权，Foo对象的retaincount变成1。

3. 微信公众平台向特定用户推送消息
4. 圆角与半角互相转换
5. json格式化和查看工具
6. 程序人生之办公桌
7. SSL协议详解
8. 推荐近期阅读过的几本书
9. 图解HTTPS
10. 招聘.net开发工程师

我的标签

WF4.0(45)
超级简单(26)
ASP.NET MVC(13)
WPF(10)
WCF(5)
音频聊天室(4)
序列化(3)
小技巧(2)
asp.net(2)
偷闲搞笑(2)
更多

随笔档案(269)

2013年12月 (1)
2013年10月 (2)
2013年8月 (1)
2013年2月 (1)
2012年12月 (1)
2012年10月 (1)
2012年8月 (1)
2012年7月 (2)
2012年6月 (2)
2012年5月 (1)
2012年4月 (2)

```
[myFooOne release];
```

但myFooTwo不在需要的时候，同样通过调用release方法，解除与Foo对象的拥有权，Foo对象的retaincount变成0。

内存泄露

我们经常会在一个方法中声明对象，看下面这个例子：

```
-(void) myMethod {
//incorrect method
NSString * myString = [[NSString alloc] init]; //retainCount = 1
Foo * myFoo = [[Foo alloc] initWithName:myString]; //retainCount = 1
NSLog(@"Foo's Name:%@", [myFoo getName]);
}
```

这上面这个方法中，我们为myString 和myFoo分配了内存空间。方法执行结束之后，两个变量超出了作用域的范围，所以不再有效。但是这个方法并没有releases这两个对象。所以运行时没有释放这两个变量占据的内存空间。除非你的应用程序结束，否则这两个变量占据的内存空间一直都是不可用的。我们把它称之为内存泄露。

为了防止内存泄露。无论什么时候，我们创建一个对象，或者创建一个对象的拷贝，我们都必须通过release方法释放。

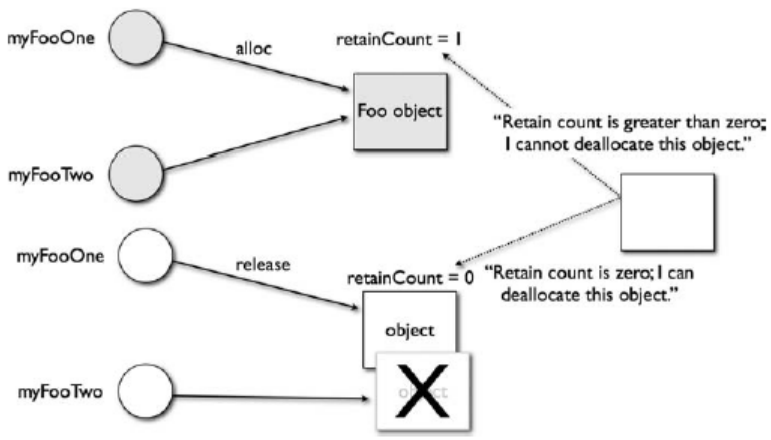
```
-(void) myMethod {
NSString * myString = [[NSString alloc] init]; //retainCount=1
Foo * myFoo = [[Foo alloc] initWithName:myString]; //retainCount=1
NSLog("Foo's Name:%@", [myFoo getName]);
[myFoo release]; //retainCount=0 so deallocate
[myString release]; //retainCount=0 so deallocate
}
```

弱引用

看下面的例子：

```
-(void) myMethod {
//an incorrect method
Foo * myFooOne = [[Foo alloc] initWithName:@"James"]; //retainCount=1
Foo * myFooTwo = myFooOne; //retainCount still 1
[myFooOne release]; //retaincount=0 so deallocated
NSLog("Name:%@", [myFooTwo printOutName]); //runtime error
}
```

nyFooTwo指向了Foo对象，但是没有调用retain方法，就是一种弱引用，上面的代码会在运行时报错。因为myFooOne调用release方法。retaincount变成0，运行时，回收了对象的内存空间。然后myFooTwo调用printPutName自然就报错了，见下图说明。



总结：本文简单的介绍了一下手动的进行内存管理、内存泄露、弱引用等objective-c的知识。