

# 疯狂の小石子

走自己的路，让别人说去吧！

博客园 闪存 首页 新随笔 联系 管理 订阅 XML

随笔- 55 文章- 4 评论- 142

## 【转】iOS中属性与成员变量的区别

【转载自并整理 <http://blog.csdn.net/itianyj/article/details/8618128>】

### 一、类Class中的属性property

在ios第一版中，我们为输出口同时声明了属性和底层实例变量，那时，属性是oc语言的一个新的机制，并且要求你必须声明与之对应的实例变量，例如：

```
@interface MyViewController :UIViewController
{
    UIButton *myButton;
}
@property (nonatomic, retain) UIButton *myButton;
@end
```

最近，苹果将默认编译器从GCC转换为LLVM(low level virtual machine)，从此不再需要为属性声明实例变量了。如果LLVM发现一个没有匹配实例变量的属性，它将自动创建一个以下划线开头的实例变量。因此，在这个版本中，我们不再为输出口声明实例变量。

例如：MyViewController.h文件

```
@interface MyViewController :UIViewController
@property (nonatomic, retain) UIButton *myButton;
@end
```

在MyViewController.m文件中，**编译器也会自动的生成一个实例变量\_myButton**。那么在.m文件中可以直接的使用\_myButton实例变量，也可以通过属性self.myButton.都是一样的。

注意**这里的self.myButton其实是调用的myButton属性的getter/setter方法**。这与C++中点的使用是有区别的，C++中的点可以直接访问成员变量(也就是实例变量)。

例如在oc中有如下代码

.h文件

```
@interface MyViewController :UIViewController
{
    NSString *name;
}
@end
```

.m文件中，self.name 这样的表达式是错误的。xcode会提示你使用->,改成self->name就可以了。因为oc中点表达式是表示调用方法，而上面的代码中没有name这个方法。

oc语法关于点表达式的说明："点表达式(.)看起来与C语言中的结构体访问以及java语言汇总的对象访问有点类似，其实这是oc的设计人员有意为之。如果点表达式出现在等号 = 左边，该属性名称的setter方法将被调用。如果点表达式出现在右边，该属性名称的getter方法将被调用。"

所以在oc中点表达式其实就是调用对象的setter和getter方法的一种快捷方式, 例如：dealie.blah = greeble 完全等价于 [dealie.blah setBlah:greeble];

以前的用法，声明属性跟与之对应的实例变量：

```
@interface MyViewController :UITViewControlle
{
    UIButton *myButton;
}
@property (nonatomic, retain) UIButton *myButton;
@end
```

这种方法基本上使用最多，现在大部分也是在使用，因为很多开源的代码都是这种方式。但是ios5更新之后，苹果是建议以以下的方式来使用：

博主邮箱：crazypebble.android@gmail.com 欢迎各位来信交流~ 文章欢迎转载，但请保留原出处。 <http://crazypebble.cnblogs.com/>

昵称：疯狂の小石子  
园龄：5年1个月  
粉丝：78  
关注：19  
[+加关注](#)

< 2016年4月 >						
日	一	二	三	四	五	六
27	28	29	30	31	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
1	2	3	4	5	6	7

## 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](#)

## 我的标签

[iOS\(9\)](#)  
[Object-C\(4\)](#)  
[Android\(4\)](#)  
[Windows\(3\)](#)  
[NSProxy\(2\)](#)  
[CocoaPods\(2\)](#)  
[iOS TableView\(2\)](#)  
[iOS UIView\(1\)](#)  
[iOS 自有桥接\(1\)](#)  
[iOS7\(1\)](#)  
[更多](#)

## 随笔档案

[2016年3月 \(1\)](#)  
[2015年12月 \(2\)](#)  
[2014年5月 \(1\)](#)  
[2014年4月 \(2\)](#)  
[2014年3月 \(5\)](#)  
[2014年1月 \(2\)](#)  
[2013年11月 \(4\)](#)  
[2013年10月 \(1\)](#)  
[2013年9月 \(1\)](#)  
[2013年8月 \(1\)](#)

```
@interface MyViewController :UIViewController
@property (nonatomic, retain) UIButton *myButton;
@end
```

因为编译器会自动为你生成以下划线开头的实例变量\_myButton，不需要自己手动再去写实例变量。而且也不需要你在.m文件中写@synthesize myButton；也会自动为你生成setter，getter方法。**@synthesize的作用就是让编译器为你自动生成setter与getter方法。**

@synthesize 还有一个作用，可以指定与属性对应的实例变量，例如@synthesize myButton = xxx；那么self.myButton其实是操作的实例变量xxx，而不是\_myButton了。

在实际的项目中，我们一般这么写.m文件

```
@synthesize myButton;
```

这样写了之后，那么编译器会自动生成myButton的实例变量，以及相应的getter和setter方法。**注意：\_myButton这个实例变量是不存在的，因为自动生成的实例变量为myButton而不是\_myButton，所以现在@synthesize的作用就相当于指定实例变量；**

如果.m文件中写了@synthesize myButton;那么生成的实例变量就是myButton；如果没写@synthesize myButton;那么生成的实例变量就是\_myButton。所以跟以前的用法还是有点细微的区别。

## 二、类别中的属性property

类与类别中添加的属性要区分开来，因为**类别中只能添加方法，不能添加实例变量**。经常会在ios的代码中看到在类别中添加属性，这种情况下，是不会自动生成实例变量的。比如在： UINavigationController.h文件中会对UIViewController类进行扩展

```
@interface UINavigationController (UINavigationControllerItem)
@property(n nonatomic, readonly, retain) UINavigationControllerItem *navigationItem;
@property(n nonatomic) BOOL hidesBottomBarWhenPushed;
@property(n nonatomic, readonly, retain) UINavigationController *navigationController;
@end
```

这里添加的属性，不会自动生成实例变量，这里添加的属性其实是添加的getter与setter方法。

注意一点，**匿名类别(匿名扩展)是可以添加实例变量的，非匿名类别是不能添加实例变量的**，只能添加方法，或者属性（其实也是方法）。

标签: [iOS](#), [property](#)

好文要顶

关注我

收藏该文



疯狂の小石子

关注 - 19

粉丝 - 78

[+加关注](#)

2

0

(请您对文章做出评价)

« 上一篇: [在iOS7中修改键盘Return键的类型](#)

» 下一篇: [\[转\]\(译\)KVO的内部实现](#)

posted @ 2013-11-23 18:05 疯狂の小石子 阅读(4233) 评论(2) 编辑 收藏

### 评论

#1楼 2015-07-06 16:17 | 追月之羽

恩，讲解的很清晰的，多谢。

支持(1) 反对(0)

#2楼 2016-02-18 15:21 | 米幽

这篇博客是目前我看到的总结的最清晰的，关于属性与下划线的描述清晰易懂。大赞

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云 - 专注为 App 开发者提供IM云服务

2013年6月 (3)

2013年5月 (3)

2013年4月 (2)

2013年3月 (1)

2013年1月 (1)

2012年9月 (2)

2012年6月 (1)

2012年4月 (1)

2011年7月 (1)

2011年5月 (2)

2011年4月 (12)

2011年3月 (5)

2011年2月 (1)

## 相册

我的头像(1)

## 技术

[eclipse中集成android ndk开发环境的配置](#)

## 最新评论

1. Re:【转】iOS中属性与成员变量的区别  
这篇博客是目前我看到的总结的最清晰的，关于属性与下划线的描述清晰易懂。大赞  
--米幽

2. Re:【转】iOS中属性与成员变量的区别  
恩，讲解的很清晰的，多谢。  
--追月之羽

## 阅读排行榜

1. 【原】Java反射机制的原理及在Android下的简单应用(13585)
2. Android下结束进程的方法(11318)
3. 在Android中使用Handler和Thread线程执行后台操作(11001)
4. Android平台下实现一个进程管理器(11000)
5. 【原创】枚举Android系统的进程、任务和服务的信息(8419)

## 评论排行榜

1. Android平台下实现一个进程管理器(74)
2. Win7下搭建外网环境的SVN服务器(18)
3. 【转】Windows平台下Android源码的下载(8)
4. 【原】Java反射机制的原理及在Android下的简单应用(8)
5. Android SDK 版本号与版本名称对照及一个编程小技巧(7)

## 推荐排行榜

1. Android平台下实现一个进程管理器(10)
2. 【原】Java反射机制的原理及在Android下的简单应用(8)
3. Android下加载Gif动画的类库(6)
4. Win7下搭建外网环境的SVN服务器(6)
5. Android 源代码结构(6)