首页 资讯 精华 论坛 回答 博客 专栏 群组 更多 ▼ 您还未登录!登录 注册

Just Code

- 博客
- 微博
- 相册
- 收藏
- 留言
- <u>关于我</u>



Objective-C中的@property和@synthesize用法

博客分类:

• Phone / IOS / Objective-C / Swift

@代表"Objective-C"的标志,证明您正在使用Objective-C语言

Objective-C语言关键词,@property与@synthesize配对使用。

功能: 让编译好器自动编写一个与数据成员同名的方法声明来省去读写方法的声明。

如:

1、在头文件中:

C代码 🗐 😭

1. @property int count;

等效于在头文件中声明2个方法:

C代码 🗐 😭

- 1. (**int**)count;
- 2. -(void)setCount:(int)newCount;

2、实现文件(.m)中

C代码 🗐 😭

1. @synthesize count;

等效于在实现文件(.m)中实现2个方法。

C代码 🗐 😭

8. }

- (int)count
 {
 return count;
 }
 -(void)setCount:(int)newCount
 {
 count = newCount;
- 以上等效的函数部分由编译器自动帮开发者填充完成,简化了编码输入工作量。

格式:

声明property的语法为: @property (参数1,参数2) 类型 名字;

如:

C代码 🗐 ☆

1. @property(nonatomic,retain) UIWindow *window;

其中参数主要分为三类:

读写属性: (readwrite/readonly)

setter语意: (assign/retain/copy)

原子性: (atomicity/nonatomic)

各参数意义如下:

readwrite: 产生setter\getter方法

readonly: 只产生简单的getter,没有setter。

assign: 默认类型,setter方法直接赋值,而不进行retain操作

retain: setter方法对参数进行release旧值,再retain新值。

copy: setter方法进行Copy操作,与retain一样

nonatomic: 禁止多线程, 变量保护, 提高性能

参数类型

参数中比较复杂的是retain和copy, 具体分析如下:

getter 分析

1,

C代码 🗐 😭

- 1. @property(nonatomic,retain)test* thetest;
- 2. @property(nonatomic ,copy)test* thetest;

等效代码:

C代码 🗐 😭

- 1. -(void)thetest
- 2. {
- 3. return thetest;
- 4. }

2,

C代码 🗐 😭

- 1. @property(retain)test* thetest;
- 2. @property(copy)test* thetest;

等效代码:

C代码 🗐 😭

- 1. -(void)thetest
- 2. {
- 3. [thetest retain];
- 4. return [thetest autorelease];
- 5. }

setter分析

1,

C代码 🗐 🌣

- 1. @property(nonatomic,retain)test* thetest;
- 2. @property(retain)test* thetest;

等效于:

C代码 🗐 😭

```
 -(void)setThetest:(test *)newThetest {
 if (thetest!= newThetest) {
 [thetestrelease];
 thetest= [newThetest retain];
 }
```

2,

C代码 □ ☆

- 1. @property(nonatomic,copy)test* thetest;
- 2. @property(copy)test* thetest;

等效于:

C代码 🥛 🔯

```
 -(void)setThetest:(test *)newThetest {
 if (thetest!= newThetest) {
 [thetest release];
 thetest= [newThetest copy];
 }
```

nonatomic

如果使用多线程,有时会出现两个线程互相等待对方导致锁死的情况(具体可以搜下线程方面的注意事项去了解)。在没有(nonatomic)的情况下,即默认(atomic),会防止这种线程互斥出现,但是会消耗一定的资源。所以如果不是多线程的程序,打上(nonatomic)即可

retain

代码说明

如果只是@property NSString*str; 则通过@synthesize自动生成的setter代码为:

C代码 🗐 😭

-(void)setStr:(NSString*)value{
 str=value;
 }

如果是@property(retain)NSString*str; 则自动的setter内容为:

C代码 🗐 😭

```
 -(void)setStr:(NSString*)v{
 if(v!=str){
 [str release];
 str=[v retain];
 }
```

所有者属性

我们先来看看与所有权有关系的属性,关键字间的对应关系。

属性值 关键字 所有权

strong	strong	有
weak	weak	无
unsafe_unretained	unsafe_unretained	无
сору	strong	有
assign	unsafe_unretained	无
retain	strong	有

strong

该属性值对应 __strong 关键字,即该属性所声明的变量将成为对象的持有者。

weak

该属性对应 __weak 关键字,与 __weak 定义的变量一致,该属性所声明的变量将没有对象的所有权,并且当对象被破弃之后,对象将被自动赋值nil。

并且,delegate 和 Outlet 应该用 weak 属性来声明。同时,如上一回介绍的 iOS 5 之前的版本是没有 __weak 关键字的,所以 weak 属性是不能使用的。这种情况我们使用 unsafe_unretained。

unsafe unretained

等效于__unsafe_unretaind关键字声明的变量;像上面说明的,iOS 5之前的系统用该属

性代替 weak 来使用。

copy

与 strong 的区别是声明变量是拷贝对象的持有者。

<u>assign</u>

一般Scalar Varible用该属性声明,比如,int, BOOL。

retain

该属性与 strong 一致;只是可读性更强一些。

参考:

http://blog.eddie.com.tw/2010/12/08/property-and-synthesize/

http://www.cocoachina.com/bbs/read.php?tid=7322

http://www.cnblogs.com/pinping/archive/2011/08/03/2126150.html

声明的分类

在 Objective-C官方文档 中的Property一章里有对类Property详细说明。 @property中的声明列表已分类为以下几种:

- 1, 声明属性的访问方法:
 - getter=getterName
 - setter=setterName 声明访问属性的设置与获取方法名。
- 2, 声明属性写操作权限:
 - · readwrite

声明此属性为读写属性,即可以访问设置方法(setter),也可以访问获取方法(getter),与readonly互斥。

- *readonly* 声明此属性为只读属性,只能访问此属性对应的获取方法(getter),与readwrite互斥。
- 3,声明写方法的实现:
 - **assign** 声明在setter方法中,采用直接赋值来实现设值操作。如:

C代码 ☆

- -(void)setName:(NSString*)_name{
 name = _name;
- 3. }
- retain

声明在setter方法中,需要对设过来的值进行retain 加1操作。如:

C代码 ☆

- 1. -(void)setName:(NSString*)_name{
- 2. //首先判断是否与旧对象一致,如果不一致进行赋值。
- 3. //因为如果是一个对象的话,进行if内的代码会造成一个极端的情况: 当此name的retain为1时,使此次的set操作让实例name提前释放,而达不到赋值目的。
- 4. if (name != name)
- 5. [name release];
- 6. name = [_name retain];
- 7. }
- 8. }
- copy

调用此实例的copy方法,设置克隆后的对象。实现参考retain。

- 4, 访问方法的原子性:
 - nonatomic

在默认的情况下,通过synthesized 实现的 setter与getter 都是原子性访问的。多线程同时访问时,保障访问方法同时只被访问一个线程访问,如:

- C代码 ☆
 - 1. [internal lock]; // lock using an object-level lock
 - 2. id result = [[value retain] autorelease];
 - 3. [internal unlock];
 - 4. return result;
- 但如果设置nonatomic时,属性的访问为非原子性访问。

来源: http://wiki.magiche.net/pages/viewpage.action?pageId=1540101

@synthesize tabBarController=_tabBarController;

@synthesize 中可以定义 与变量名不相同的getter和setter的命名,籍此来保护变量不会被不恰当的访问

分享到: 🔯 🔑

规范化-数据库设计原则 I 传送数据 Passing Data Between Views (实 ...

- 2012-03-07 23:58
- 浏览 59419
- 评论(0)
- 分类: 移动开发
- 相关推荐

评论

发表评论



您还没有登录,请您登录后再发表评论

一天梯梦的博客

天梯梦

• 浏览: 3811506 次

• 性别: 3

• 来自: 洛杉矶

● 我现在离线

最近访客 更多访客>>

Ίτογο

dylinshi126

iteye

ning2-eye



liujickson



athrun9

文章分类

- 全部博客 (1434)
- Php / Pear / Mysql / Node.js (277)
- Javascript /Jquery / Bootstrap / Web (288)
- Phone / IOS / Objective-C / Swift (133)
- <u>Ubuntu / Mac / Github / Aptana / Nginx / Shell / Linux (250)</u>
- Perl / Koha / Ruby / Markdown (7)
- Java / Jsp (12)
- Python 2 / Wxpython (25)
- Codeigniter / CakePHP (32)
- Div / Css / XML / HTML5 (83)
- WP / Joomla! / Magento / Shopify / Drupal / Moodle / Zimbra (176)
- Apache / VPN / Software (30)
- AS3.0/2.0 / Flex / Flash (45)
- <u>Smarty (6)</u>
- SEO (23)
- Google / Facebook / Pinterest / SNS (39)
- Tools (8)

社区版块

- 我的资讯(0)
- 我的论坛(16)

• 我的问答(0)

存档分类

- <u>2015-04</u> (3)
- 2015-03 (14)
- <u>2015-02</u> (30)
- 更多存档...

评论排行榜

- Swift中的问号?和感叹号!
- 开源ckplayer 网页播放器, 跨平台(html5 ...
- 跨浏览器兼容的HTML5视频音频播放器
- html5拖拽图片批量ajax无刷新进度上传

最新评论

- 益达555: 阿斯蒂芬阿斯蒂芬 跨浏览器兼容的HTML5视频音频播放器
- 益达555: [col阿萨德发射点发or=red][/color] 跨浏览器兼容的HTML5视频音频播放器
- <u>aslijiasheng</u>: mark顶