

属性,点语法

setter/getter

定义

在.h文件中使用@property声明属性

@property NSString *name

相当于在@interface中声明了两个方法 setter/getter

实现

在.m文件使用@synthesize实现属性

如: @synthesize name = _name

_name 是setter/getter方法的操作实例变量

属性

相当于在@implementation实现

```
setter方法
-(void)setName:(NSString *)name
{
    _name = name;
}
getter方法
-(NSString *)name {
    return _name;
}
```

synthesize在Xcode4.5之后版本可以省略,只使用@property声明实现即可
在默认情况下,当没有setter/getter方法时,通过@synthesize可以自动生成标准的setter/getter方法

属性和实例变量

属性的作用是生成setter以及getter方法的实现,如果方法内部操作的实例变量未定义,系统会自动生成一个_属性名的实例变量,但是生成的实例变量的可见度是私有的,子类不可访问。
一旦同时重写了setter、getter方法,并且没有实现@synthesize,@synthesize就不再生成实例变量。

属性的特点

读写性

readonly

只读状态,属性只能生成getter方法,不能生成setter方法

readwrite

读写状态,属性即生成setter方法又生成getter方法 默认的读写特性

setter=

指定属性生成的setter方法的名字

getter=

指定属性生成的getter方法的名字

原子性

atomic

原子特性, setter/getter方法在多线程访问下是绝对安全的 默认的原子特性

nonatomic

非原子特性, 仅仅是普通的setter/getter方法

声明属性时,通常使用nonatomic

assign

属性非对象类型(int,float)时使用

retain

属性是对象类型(NSString)时使用

copy

属性是对象类型并且想得到对象副本

语义特性

语义设置

assign

```
如果语义设置的关键字是assign。例如:
@property (nonatomic, assign) NSInteger age;

setter方法内部实现:
-(void)setAge:(NSInteger)age {
    _age = age;
}

getter方法内部实现:
-(NSInteger)age {
    return _age;
}
```

retain

```
如果语义设置的关键字是retain。例如:
@property (nonatomic, retain) NSString *name;

• setter方法内部实现:
-(void)setName:(NSString *)name {
    if (_name != name) {
        [_name release];
        _name = [name retain];
    }
}

• getter方法内部实现:
-(NSString *)name {
    return [[_name retain] autorelease];
}
```

copy

```
如果语义设置的关键字是copy。例如:
@property (nonatomic, copy) NSString *gender;

• setter内部实现
-(void)setGender:(NSString *)gender {
    if (_gender != gender) {
        [_gender release];
        _gender = [gender copy];
    }
}

• getter内部实现
-(NSString *)gender {
    return [[_gender retain] autorelease];
}
```

点语法

只要符合系统默认setter、getter书写格式的方法都可以使用点语法。例如:实例变量_name

```
• setter方法
[person setName:@"Lily"];
使用点语法:
person.name = @"Lily"; 注: 点语法点的属性名
```

```
• getter方法
NSString *string = [person name];
使用点语法:
NSString *string = person.name;
```

• 属性是一对getter、setter方法, 点语法是属性的另一种调用格式。

总结

- 1.属性是Objective-C的重要语法,属性是一组setter、getter方法,内部对实例变量进行操作。
- 2.属性三大特性以及之间的区别。
- 3.点语法。