

title: 为属性声明自定义特性

date: 2016-06-17 09:14

tags:

- 可写性

如果你给属性加上了readonly的特性，编译器就不会自动生成setter访问器方法，属性也就变成了只读的。

- setter支持的特性

setter方法不仅仅是简单的赋值，有这么几个修饰词是用来描述setter方法的：

assign、retain、copy和weak，用不同的修饰词表示生成的setter方法会有所不同，默认是assign，这意味着在setter方法中真的就是简单的赋值。retain表示赋值之后会调用新值的retain方法和旧值的release方法。copy表示先将值拷贝一份，然后将这个拷贝赋值给实例变量，这个修饰词只适用于实现了NSCopying协议的类型。

weak表示这个属性的值将是一个弱引用，引用和被引用者之间没有所属关系。如果系统销毁了某个属性所引用的对象，属性的值会自动地变为nil。一般情况下，如果父对象拥有子对象的引用，那么子对象就不应该再持有父对象的引用，而是应该使用弱引用的方式来引用父对象。

- 原子化

nonatomic指定访问器方法是非线程安全的，即多个线程可以同时访问这个属性。默认情况下，访问器方法都是线程安全的，即使在多线程环境中也毫无风险。然而，即使要保证在多线程环境中能正常工作，访问器也不一定必须是线程安全的。而将访问器方法指定为非线程安全的，可以让getter方法直接返回实例变量的值，效率更高，在iPhone应用中特别推荐使用。

使用alloc、retain、copy、release的原则

1. 不应该释放不属于你控制的对象。
2. 总是需要确保调用了一个对象的retain之后必须调用这个对象的release方法。
3. 通过alloc或者copy创建出来的对象以及调用了retain方法的对象都是由你控制的。通过构造器方法创建的对象不由你控制，比如通过stringWithString：这个方法创建的字符串对象。