# OC

1、Objective-C有两种注释：//单行注释 和 /\* \*/多行注释

2、#import 引入文件内容到该文件中，exactly as if the contents of the file were typed into the program at the point.

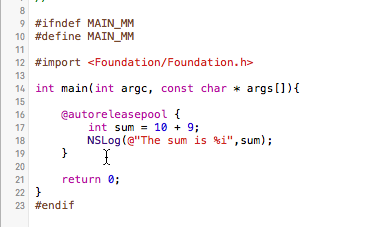
3、””是C风格字符串，@””是 NSString类字符串对象

4、格式化字符串使用 %i %s %f

5、Obkective-C使用下面的语法调用方法

[ ClassOrInstance method : parameter];

6、



7、定义一个新类使用@interface Section，并且必须指定父类

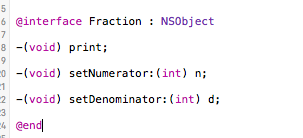
@interface NewClassName:ParentClassName

propertyAndMethodDeclarations;

//方法声明格式： -或者+(返回值) 方法名: (参数类型) 参数名;

@end

类名第一个字母大写，Instance variables, objects, and method names第一个字母小写。



-表示实例方法，+表示类方法。可以在@interface中声明和封装变量一样名字的方法，这是通用的作法。

8、@implementation定义类的实现，格式如下：

@implementation NewClassName

{

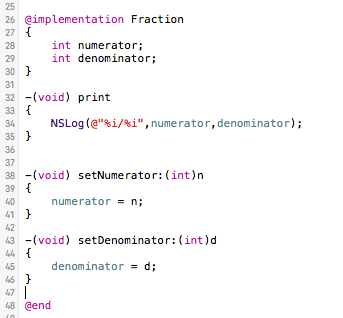
memberDeclarations;//定义存储的实例变量，变量在类外不能使用。可以在@interface中声明和封装变量一样名字的方法来提供封装变量的访问，这是通用的作法。

}

MethodDefinitions

@end

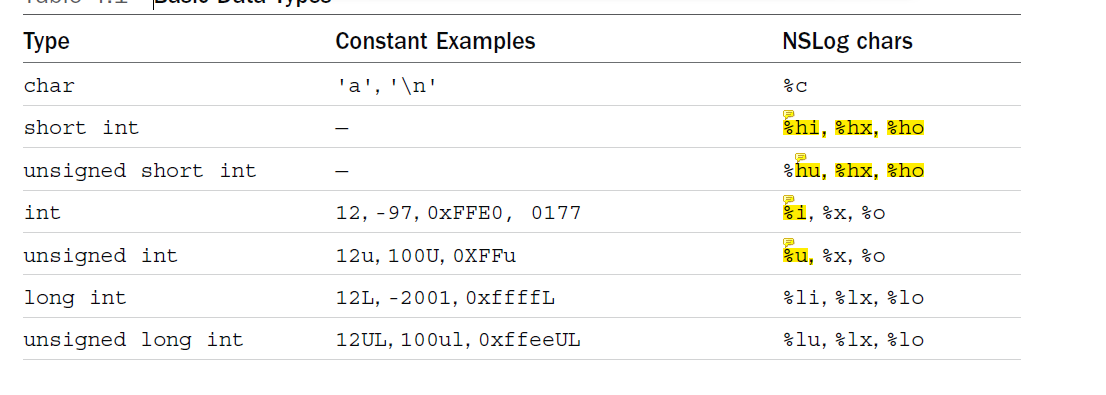
NewClassName必须和接口中的名字一样。可选的加和@interface中的父类声明 :ParentClassName

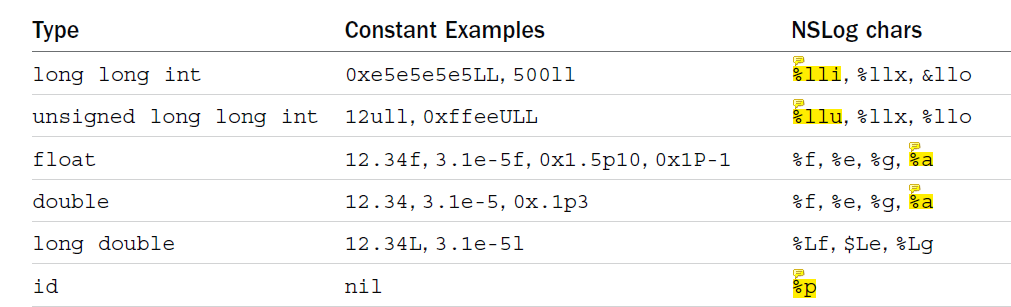


9、Fraction \*myFraction; \*代表myFraction是一个引用或者指针指向Fraction对象。

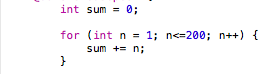
10、可以用alloc init来实例化变量，也可以用new来实例化。New是alloc和init的组合调用。

11、id数据类型可以存储任何类型的数据，它是通用数据类型。

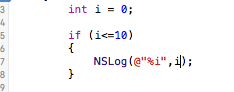




12、#import确保头文件只会被导入一次。所以不需要加#ifndef #define #endif #import导入两种方法： #import<> 和 #import”” <>表示引入系统类，会直接到库里查找，””表示引入自定义类，会优先查找当前目录，，在查找path目录，再到库里查找。

13、

14、for while do while 循环与C#、C 、Java完全一样

15、if else if else switch ！ ?:与C#、C 、Java完全一样 除了非0表示真，0表示假。

16、&& 和 || 判断操作符与C#、C 、Java完全一样，如果半边能计算出结果，不计算另半边。

17、Objective-C中的BOOL类型。使用%@来打印一个对象，因此可以打印BOOL, BOOL有YES和NO两个值。

18、大部分的C 框架中使用C99中的bool类型，所有的Objective-C框架中使用的是BOOL类型。

在obj.h中

1. **typedef** **signed** **char**     BOOL;
2. .
3. #define OBJC\_BOOL\_DEFINED
4. #define YES             (BOOL)1
5. #define NO              (BOOL)0

BOOL 是一个Objective-C 类型，signed char；BOOL 是 signed char的宏定义  
bool 是标准C类型 \_Bool 的宏定义版本，int。bool是int的宏定义版本

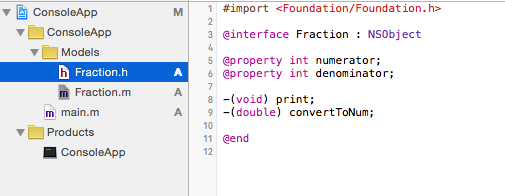
19、condition?:expression如果condition为真，返回condition，否则返回expression

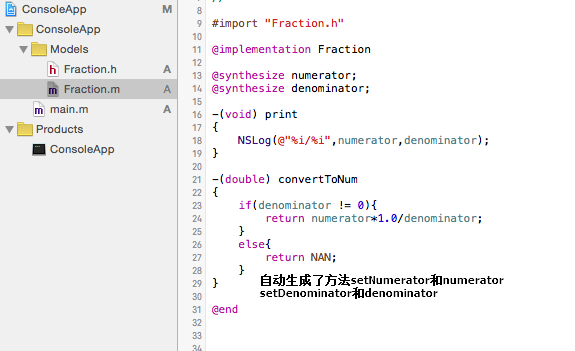
20、Objective-C没有命名空间，苹果使用两个字母前缀来命名官方库，我们使用三个或多个字母做前缀命名我们自己的类。本人使用LCM

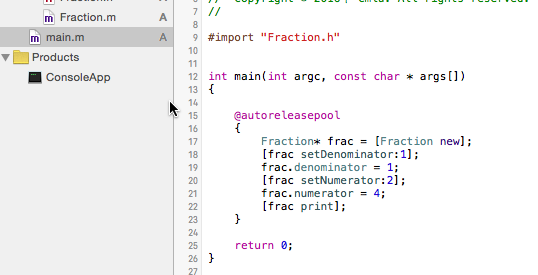
21、[类定义@interface放到Class.h](mailto:类定义@interface放到Class.h)文件中，[类实现@implementation放到Class.m](mailto:类实现@implementation放到Class.m)文件中。在Class.m中引入#import ”Class.h”，使用Class的其他程序文件，只需要引用Class.h，Class.h定义类的公共部分，Class.m包含私有部分：实例变量和实现代码。

22、@property @synthesize自动生成get和set, 以X为例生成setX和X访问方法。

@property在interface文件中声明私有实例变量，@synthesize在implementation文件中生成get和set,@synthesize可以省略，如果省略，则生成的get加\_前缀，所以不能省略。

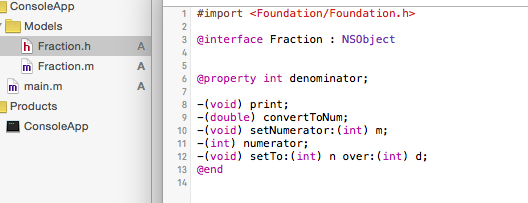


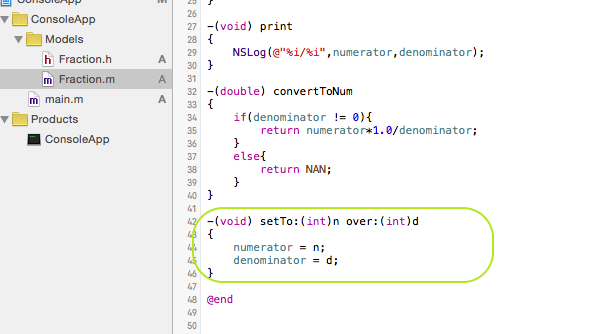


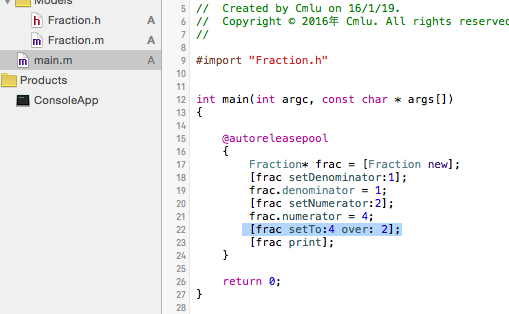


23、instance.property可以通过instance.property来访问property,instance.property=value来设置值。无论是手动定义的，还是自动生成的。也可以使用原有的[instance property]来访问，[instance setProperty:value]来设置值。

24、声明多个参数的方法：







25、实际上参数名是可选的，可以如下定义

-(int) set: (int) n : (int) d; [myFraction set:1:3];

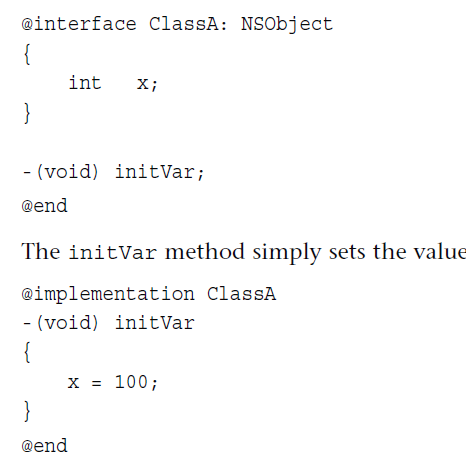
Apple不推荐这种方式

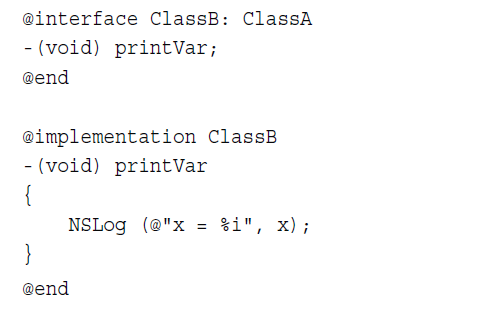
26、Class\* instance加\*类似于C#、java的类引用。

27、static int hitCount = 0;static本地变量能够在多次调用间保存，仅在第一次调用时初始化。不要使用这个特性。

28、self类似于java和C#中的this.

29、私有实例变量不能被子类访问。，只有在interface中声明的实例变量才能被子类访问，但在子类外是不能被直接访问的。在implementation中声明的变量或synthesized的不能被子类访问。

30、



31、Objective-c是单继承的。

32、#import会包含这个类的所有信息，包括实体变量和方法，而@class只是告诉编译器，其后面声明的名称是类的名称，至于这个类是如何定义的，暂时不用考虑，后面会再告诉使用者。

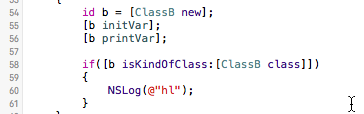
在类的声明.h文件中一般只需要知道被引用的类的名称就可以了，不需要知道其内部的实体变量和方法，所以在类的声明.h文件中一般使用@class来声明这个名称是类的名称。而在实现类里面，因为会用到这个引用类的内部的实体变量和方法，所以需要使用#import来包含这个被引用类的头文件。

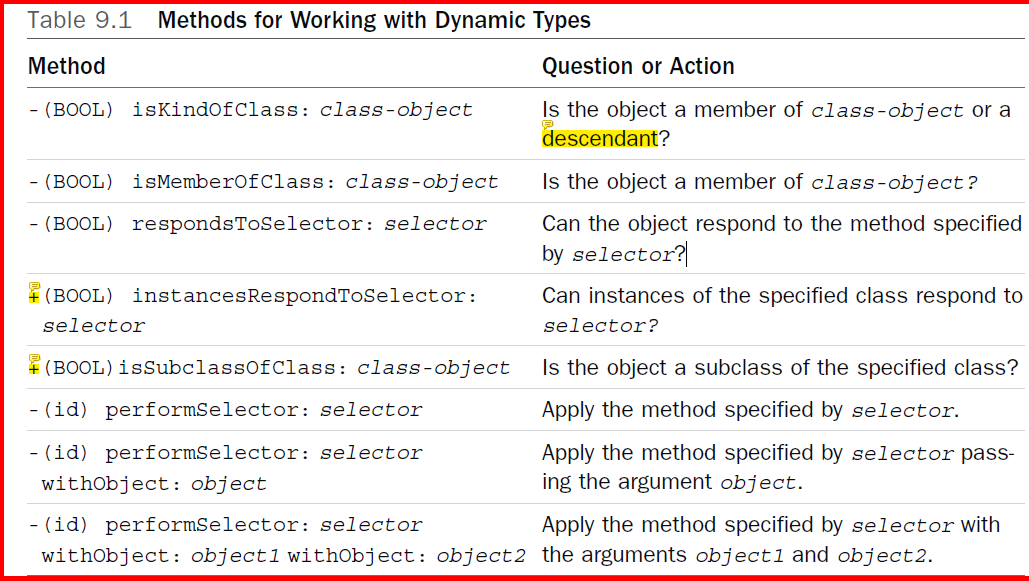
如果有循环依赖关系，如：A->B,B->A这样的相互依赖关系，如果使用#import来相互包含，那么就会出现编译错误，如果使用@class在两个类的头文件中相互声明，则不会有编译错误出现。

所以一般来说@class是放在.h文件中的，只是为了在.h中引用这个类，把这个类作为一个类型来引用的。在实现这个.h文件的.m文件中，如果需要引用这个类的实体变量或者方法之类的，还是要用#import。

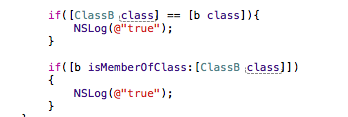
33、Override Method 在子类中定义签名相同的方法即可实现多态。

34、id是动态类型，类似于C#中的dynamic类型。Id dataValue; id类型不能使用dot operator。

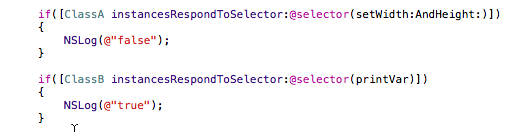
35、 isKindOfClass判断对象是否是某个类或子类对象。 isMemberOfClass判断对象是否是本类的实例。



这些方法在NSObject中定义。

class应用于类或者对象上，返回其对应的类。

使用@selector指令返回SEL 类型。





36、@try @catch @finally @throw和C#一样

@try {

<#statements#>

}

@catch (NSException \*exception) {

<#handler#>

}

@finally {

<#statements#>

}

37、在你的类中覆盖init

-(instancetype) init

{

self = [super init];

If(self){

//初始化代码

}

return self;

}

在程序运行过程中，它会在你程序中每个类调用一次initialize。这个调用的时间发生在你的类接收到消息之前，但是在它的超类接收到initialize之后。实际上initialize类似于C#中的static静态初始化

+(void) initialize

{}

38、super表示父类，self表示自身。

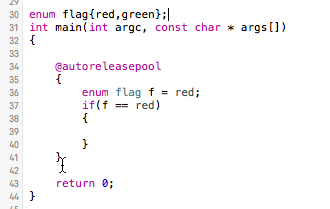
39、@synthesize window = \_window; it says to synthesize the getter and setter for the property named window and to associate that property with an instance variable called window。\_window是私有的，类外看不到。Window是property,\_window是实例变量。

40、全局变量以g开头，定义在类外。外部变量默认是external global variable. extern int gMoveNumber;//声明， 其它文件使用，使用extern声明。

41、oc不支持class variable，通常通过在implemention文件中声明类外的static变量来模拟静态类变量，通过类静态方法来访问。 Static全局变量仅能在声明的文件中访问。

42、

enum direction { up, down, left = 10, right };最后要加分号。



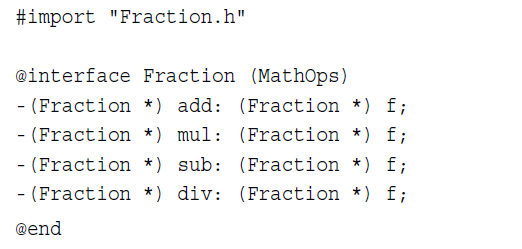
枚举本质上是int,以int对待。

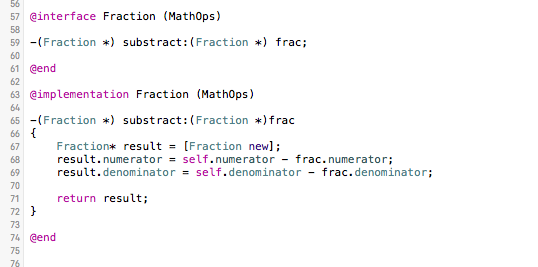
43、typedef与c一样。

44、category类似于C#中的类拓展。

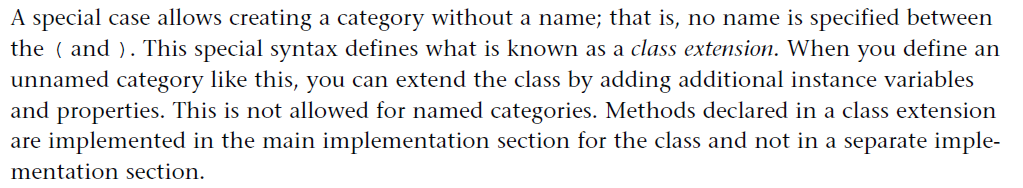
@interface 要拓展的类名 (CategoryName)

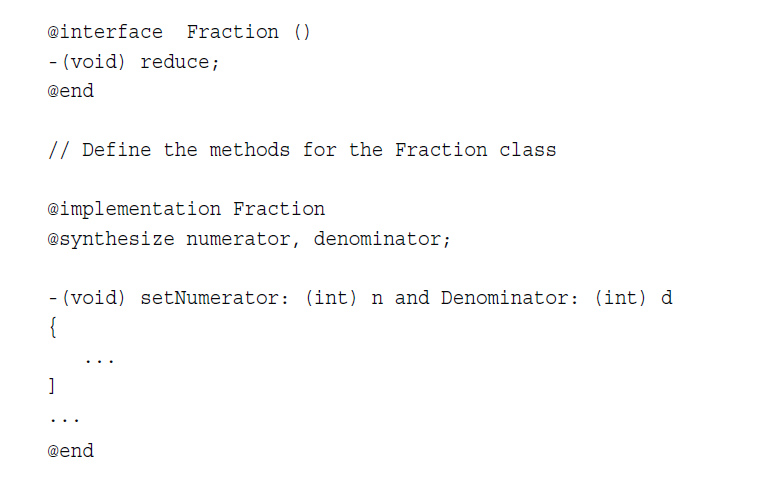
@end





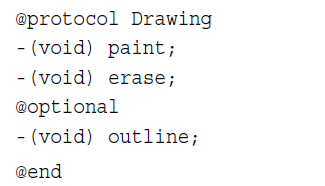
理论上，接口和实现应该放到两个文件中。

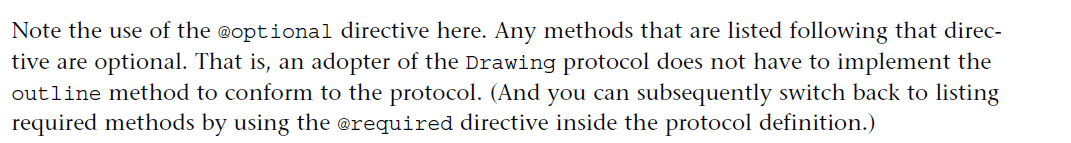


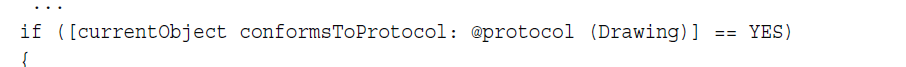


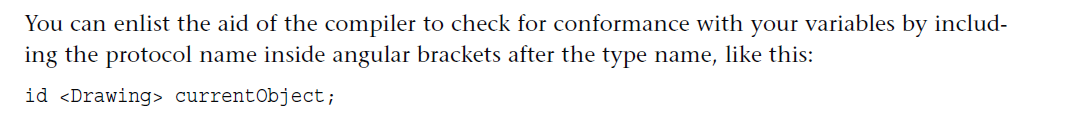
category can override another method in the class，但不要这么做。可以有任意多category。Category会传递到子类。

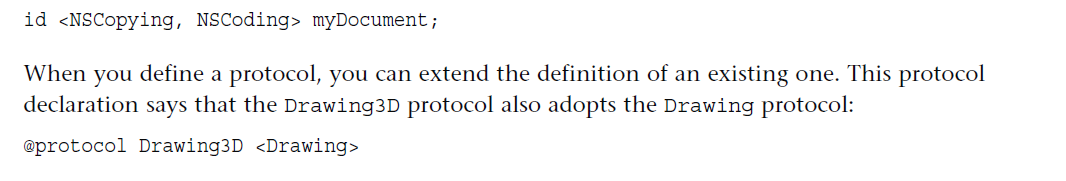
45、@protocol定义了一个方法集合，如果你的类实现了协议，在@interface父类后加<协议名> 多个协议使用<协议1,协议2> 协议类似于接口，但它可以包含实现。

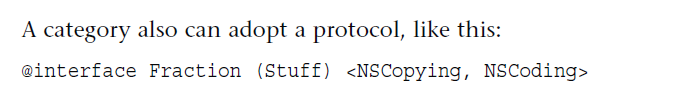


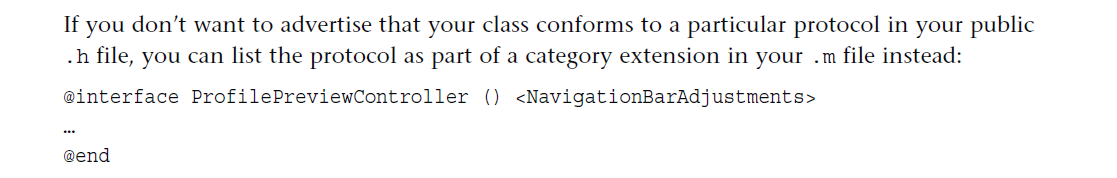


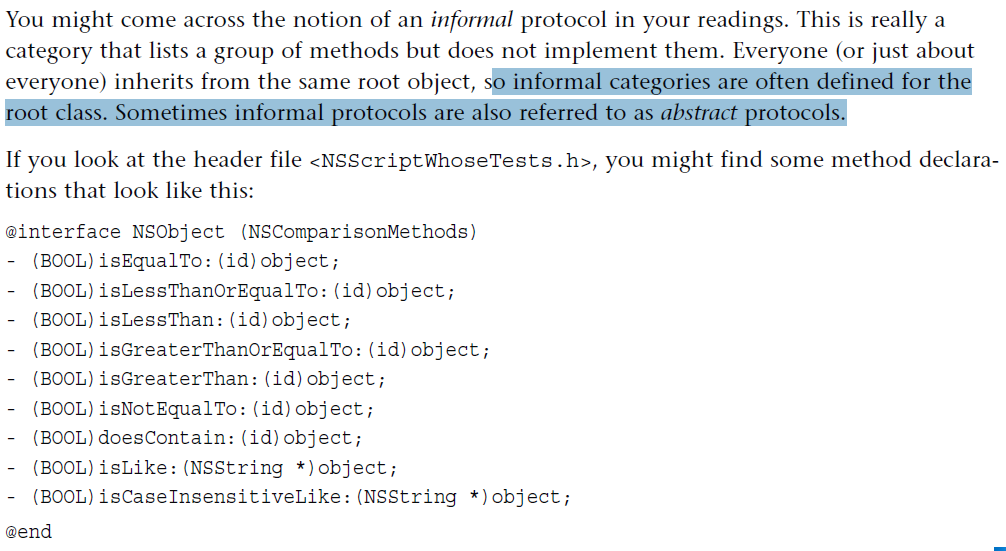


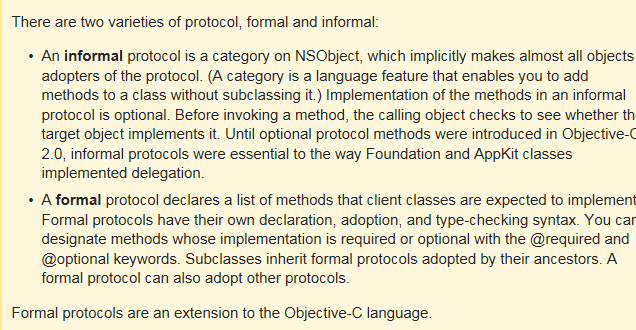


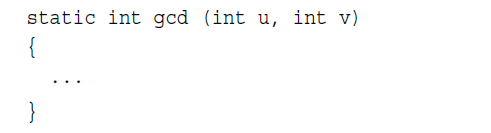






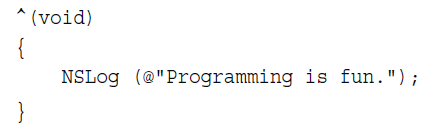
46、



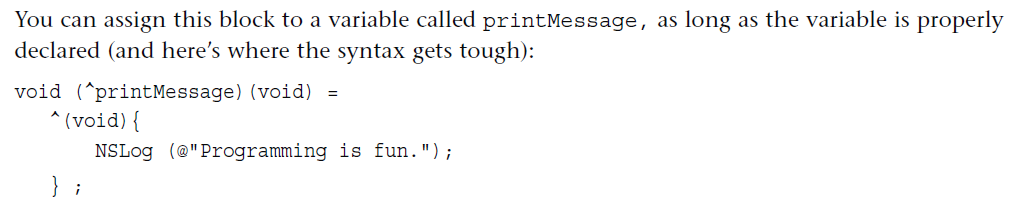
47、

函数默认是全局的，加static，只能在本文件中访问。

48、block， block可以接受参数，返回值。Block定义在函数和方法中，可以作为函数参数，可以访问block外的变量，但不能修改，除非加\_\_block标志，可以修改外边的变量。

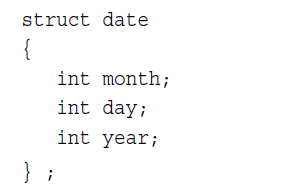


Block以^开始，后面跟着()加参数列表。



当block访问外部变量值，是在block定义时的值，而不是调用时的值。

\_ block int foo = 10;变量加了\_\_block，可以在block中修改，调用是调用时的值，不是定义时的值。

49、

struct date today; today.day = 21;

struct date today = { .month = 7, .day = 2, .year = 2014 } ;

50、int\* ptr;//声明指针 ptr = &I;//赋值， \*ptr //取ptr指针指向变量的值。 struct date\* dateptr;//指向结构date的指针。

Dateptr->day 或 (\*dateptr).day来访问成员

Fraction \*\*fractsPtr;指向指针的指针。数组名是指向数组第一个元素的指针。

C风格字符串以’\0’结尾。

int (\*fPtr) (void);函数指针 fPtr();调用函数

arrays, structures, character strings, and unions,不是object

51、

Instance variables are stored in structures。 One protected member isa in structure identifies 对象所属的Class。You can direct acces to member of an object’s structure by making them @public 。

myclass->data;

An object variable is really a pointer。如Fraction \*myFract;

Method are function, and message expressions are function calls. Objective-c compiler creates a unique name for each function using a combination of class name and function name.

The id type is a genic pointer type.

52、xcode文件组织方式，新建group，在项目中建同名文件夹，将代码文件放到该文件夹下，则可，其它文件中引用 #import “文件名”,不需要加目录

53、

Alcatraz，这个是github上的一个开源项目，用来管理Xcode上的插件。

有搜索功能，界面也很人性化。

在终端上输入如下命令即可完成安装：curl -fsSL https://raw.github.com/supermarin/Alcatraz/master/Scripts/install.sh | sh

参见文档：http://www.jianshu.com/p/5c8ed25ad434

之后安装VVDocumenter

54、nil：指向oc中对象的空指针

Nil：指向oc中类的空指针

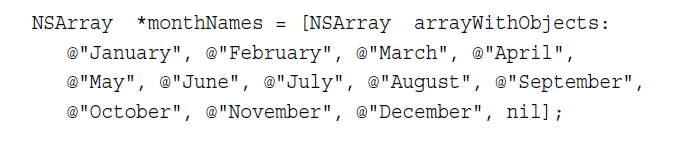
NULL：指向其他类型的空指针，如一个c类型的内存指针

NSNull：在集合对象中，表示空值的对象

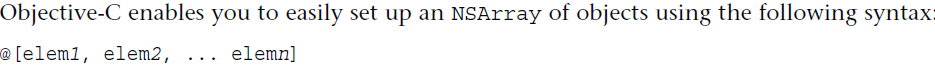
55、NSString包含unichar，unichar是一个多字节字符，符合unicode标准。%@显示object，实际上调用description方法。NSString是不可修改的。NSMutableString是可修改的，是NSString的子类。类似于C#StringBuilder.

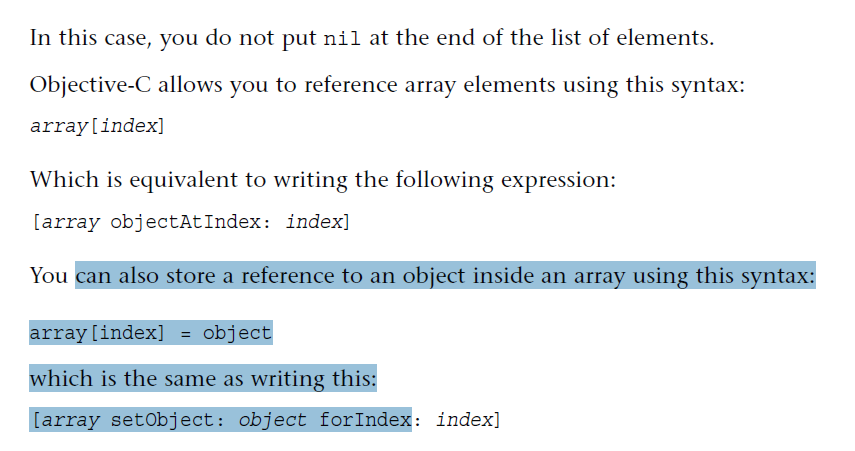
NSStringEncoding类似于C#Encoding。

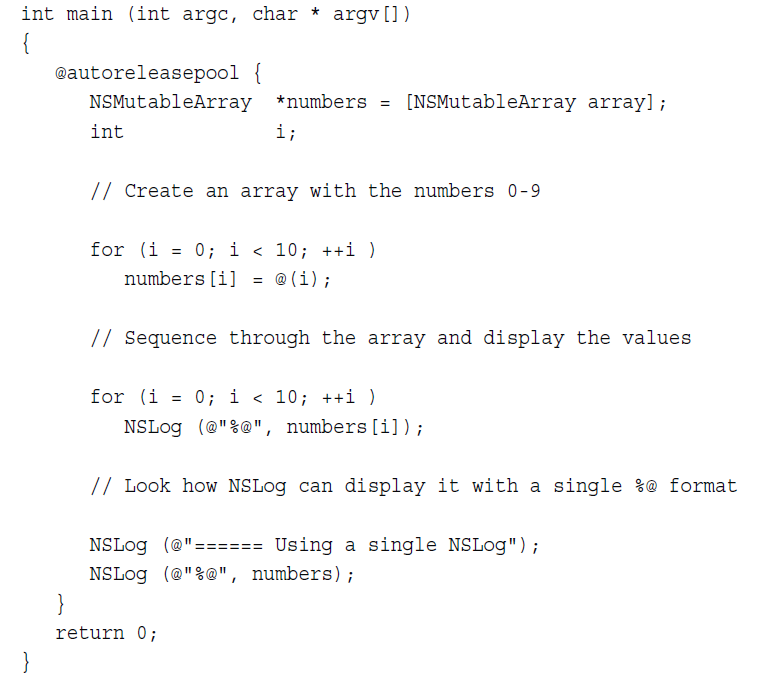
56、Array是有序的集合，mutable array NSMutableArray是Immutable NSArray的子类。



Nil必须放到list的最后一个，nil must be specified as the last value in the list; it is not actually stored inside the array.

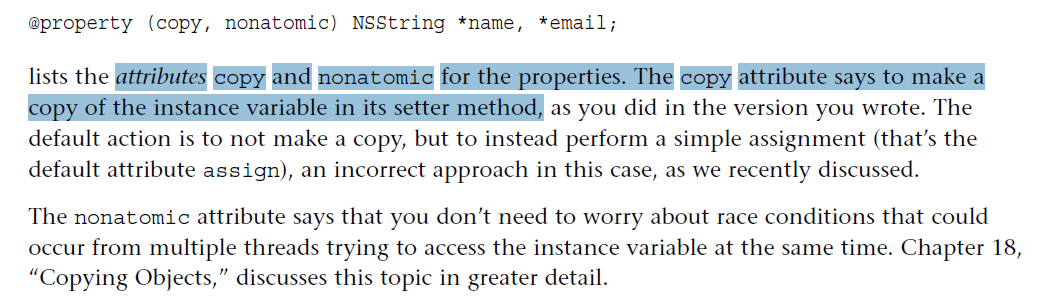
57、

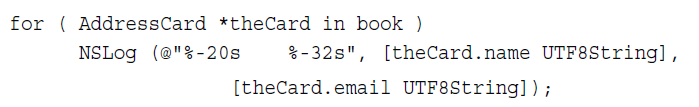


58

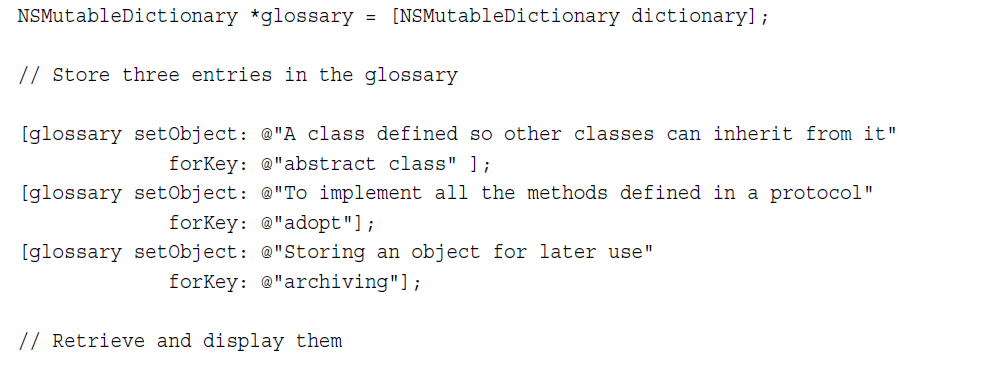
集合中只能存对象，不能存基本类型。

实例变量建议使用\_开头

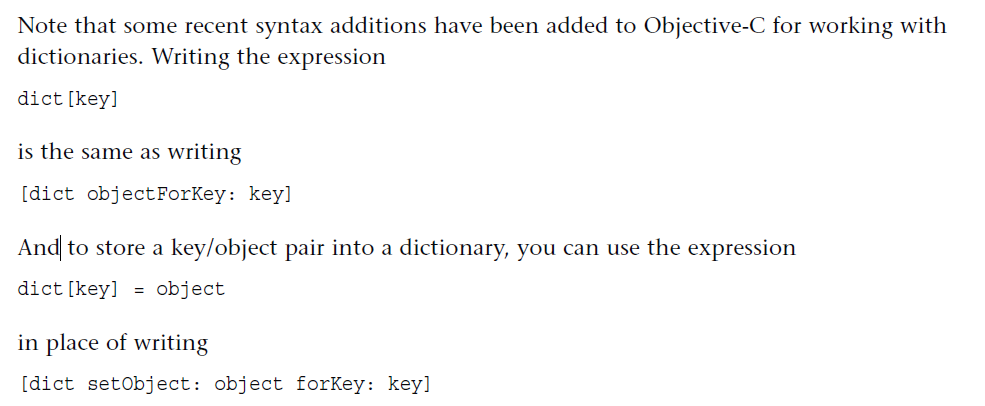
59、

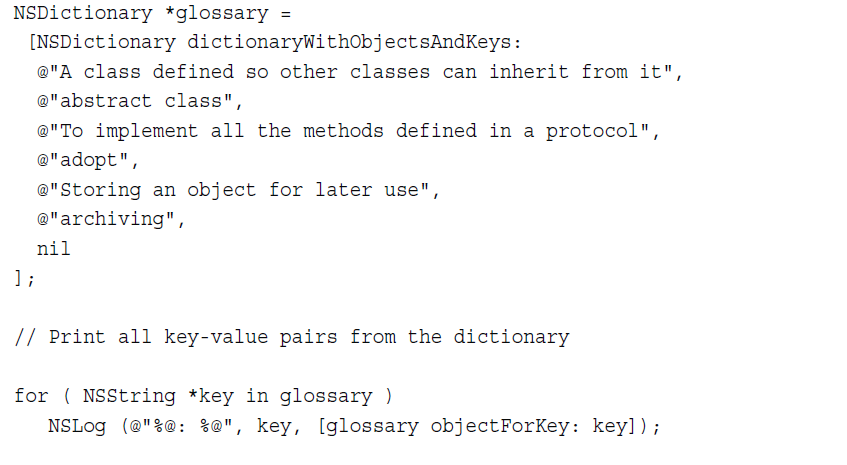


60、Dictionary的键和值都必须是object。但不能是nil.



Dictionary有可变和不可变两种：NSDictionary 和 NSMutableDictionary。





The list must be terminated with the special nil object.

61、A *set* is a collection of unique objects, and it can be mutable or immutable. NSSet , NSMutableSet , and NSIndexSet.集合有可变和不可变之分。

NSOrderedSet and NSMutableOrderedSet NSIndexSet NSIndexSet没有Mutable版本。

NSIndexSet  是个无符号整数集合。集合中的元素不可变的、不可重复。常被用来当作索引使用。就从它字面上理解，就叫做：索引集合。

NSCountedSet:计算加入集合的次数。

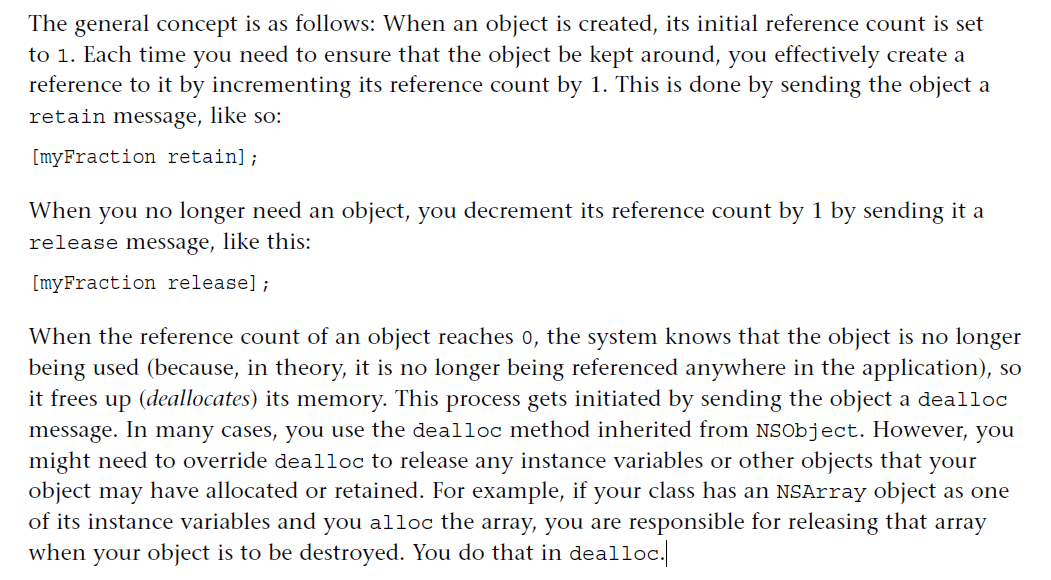
62、/根目录，~user’s home directory。

63、NSData和NSMutableData提供了面向对象封装的byte Buffer。

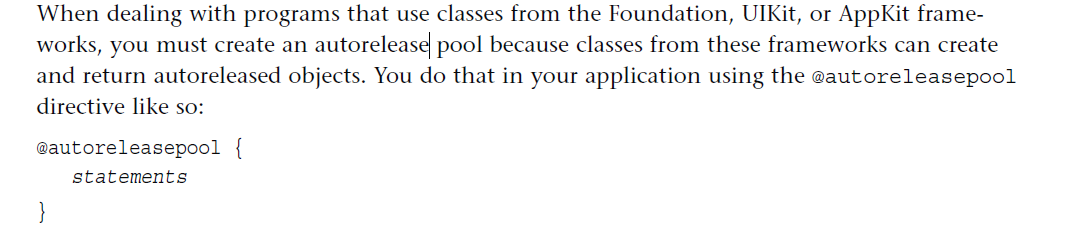
64、OC支持三种基本内存管理模型：1、Automatic garbage collection 2、Manual reference counting and the autorelease pool 3、Automatic Reference Counting(ARC)

Automatic garbage collection 系统自动决定应该释放哪些内存，但IOS不支持GC, Apple不鼓励这种方式。

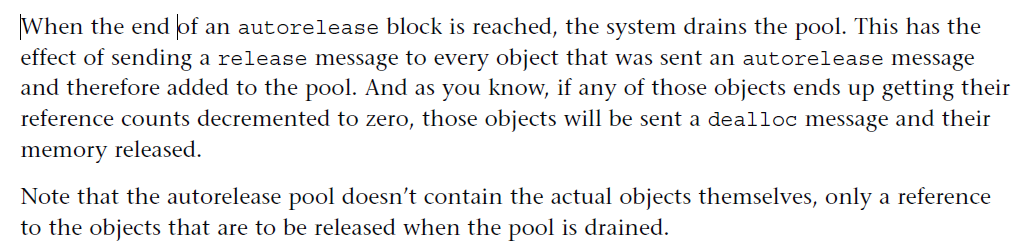
Manual reference counting and the autorelease pool:

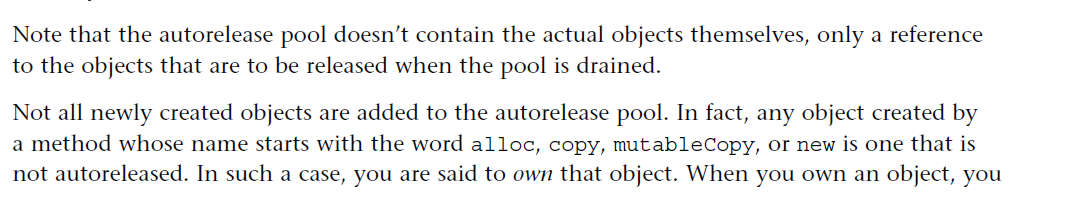


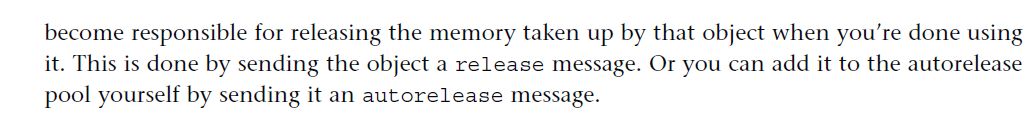
NSAutoreleasePool类用来跟踪对象to be released at a later time. 当Pool调用drain方法时释放pool跟踪的对象。 将对象加到autorealse pool,发送autorelease消息。[obj autorelease]



当}到达时，调用drain方法。

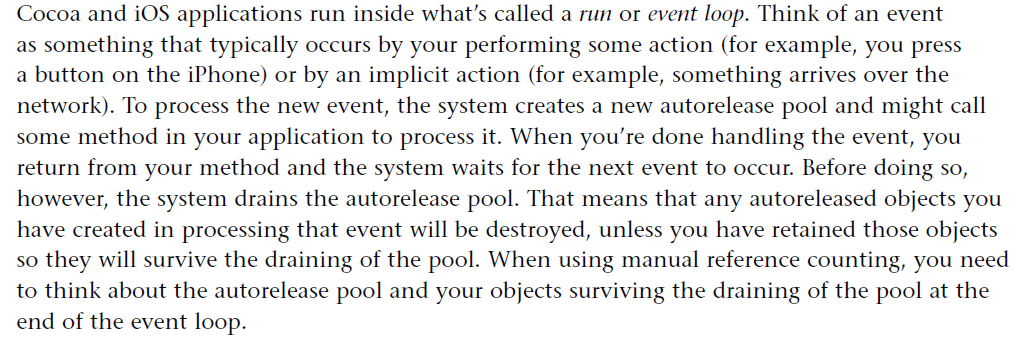


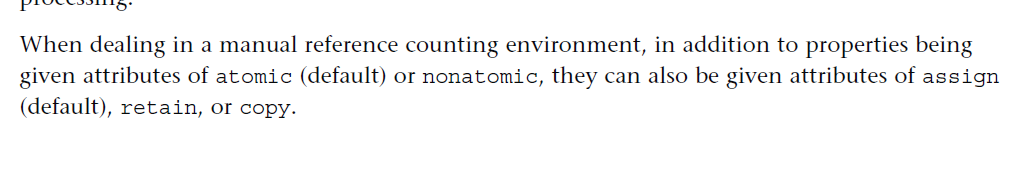


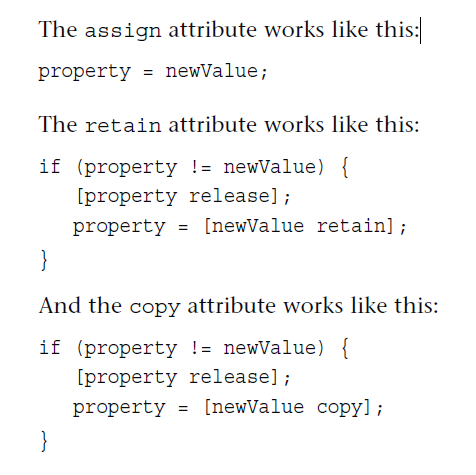


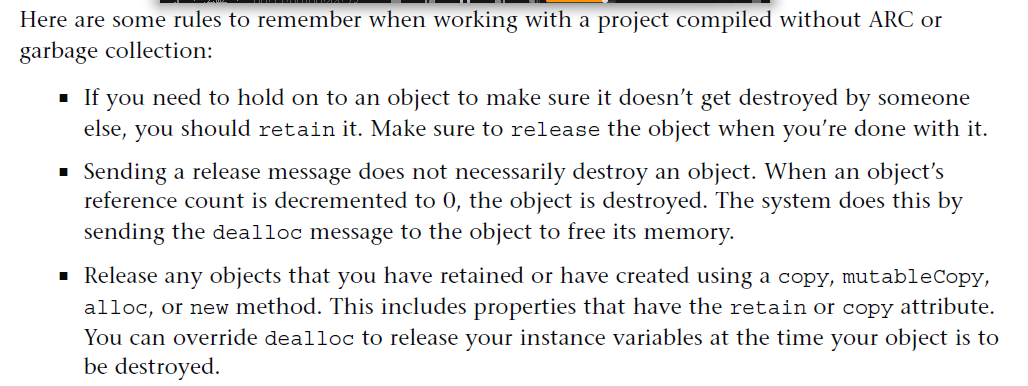
[[[Fraction alloc] init] autorelease];

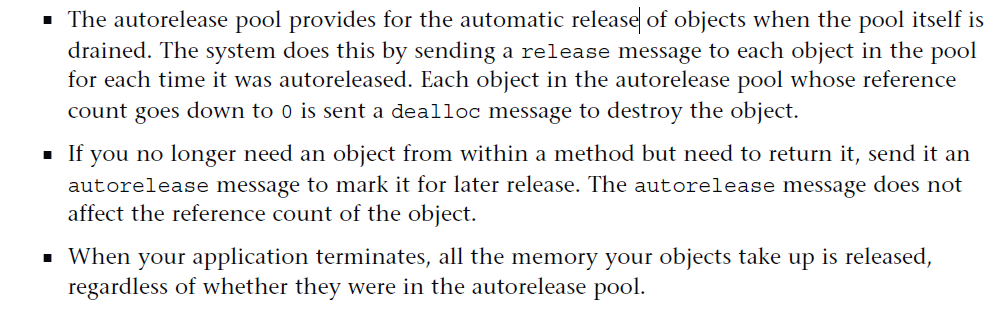
Cocoa and iOS applications run inside what’s called a *run* or *event loop.*

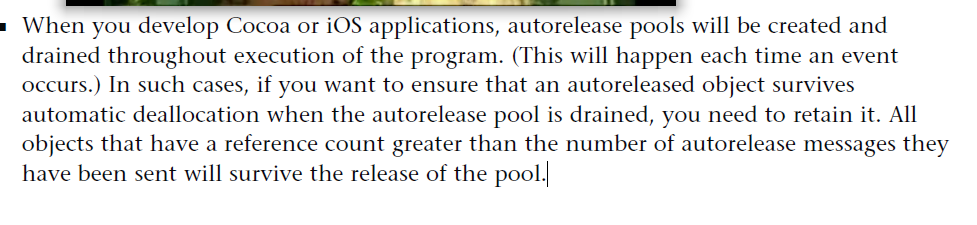


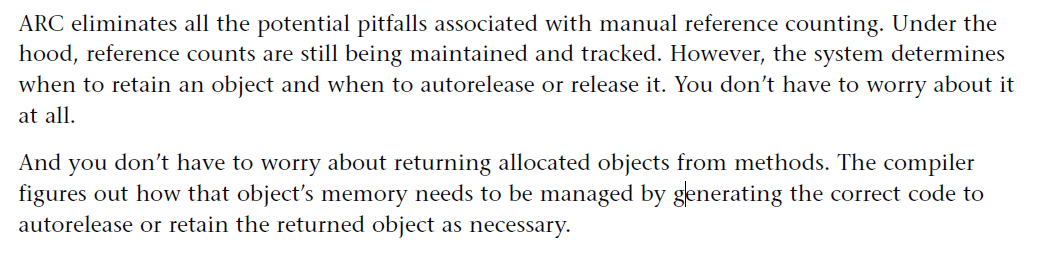




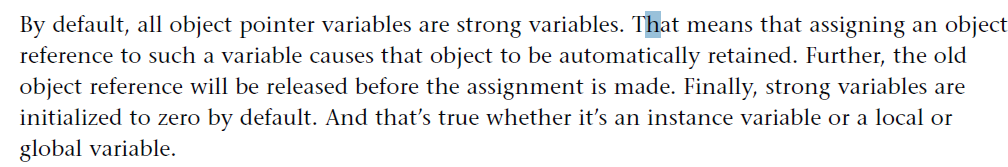


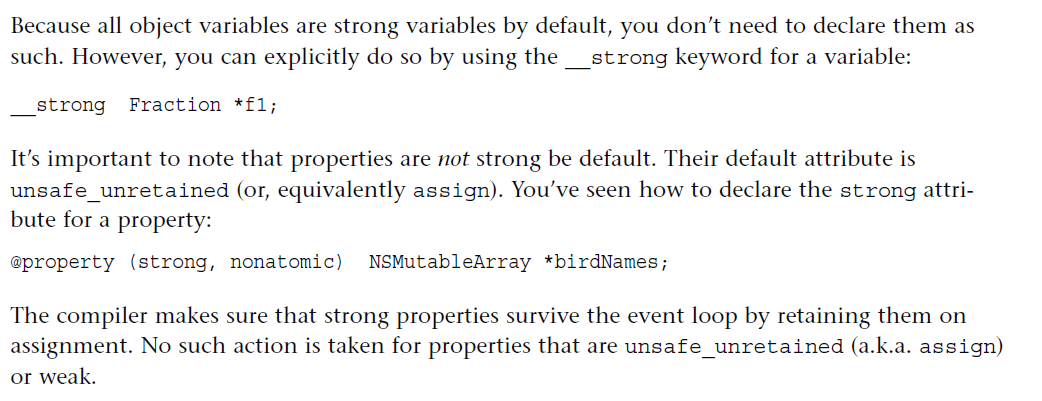




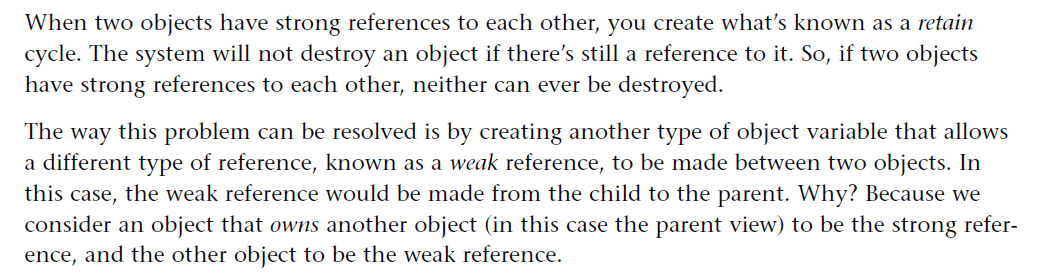


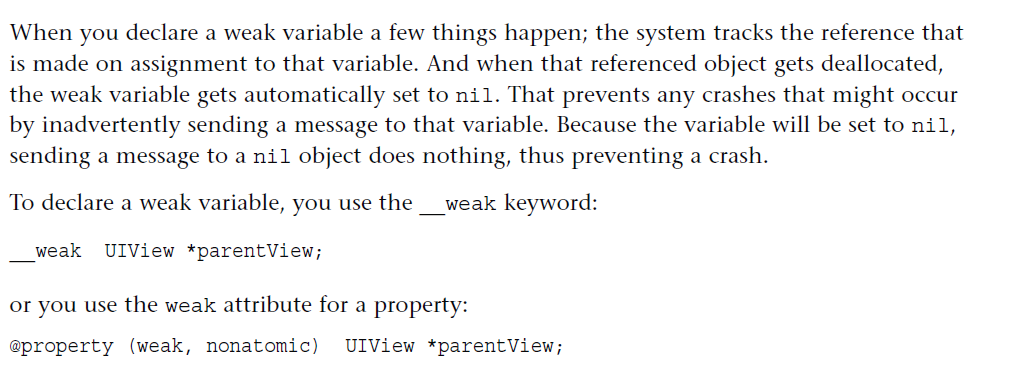
By default, all object pointer variables are strong variables.

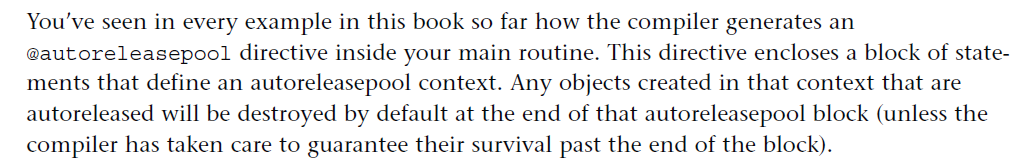


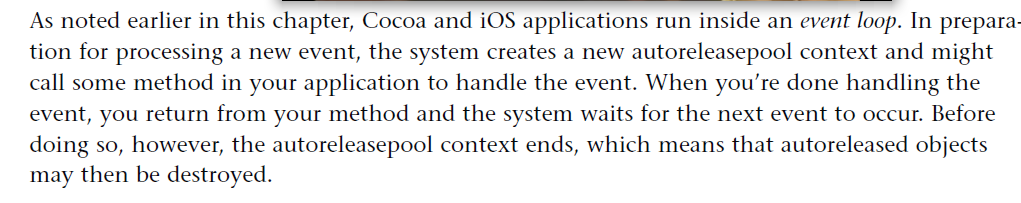


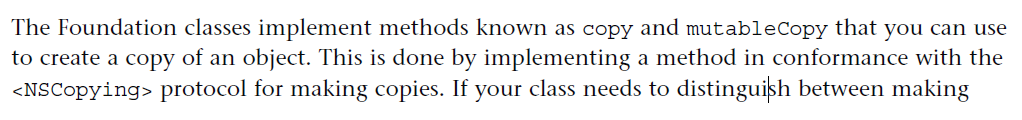
Sometimes you set up a relationship between two objects where each object needs to reference the other.

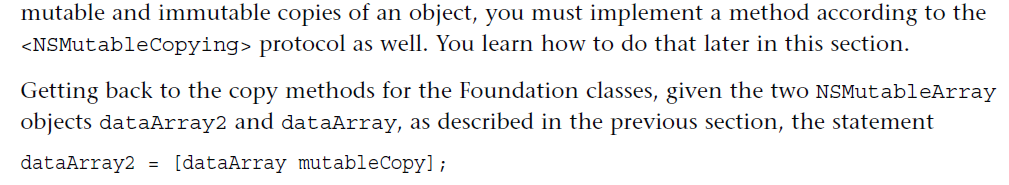




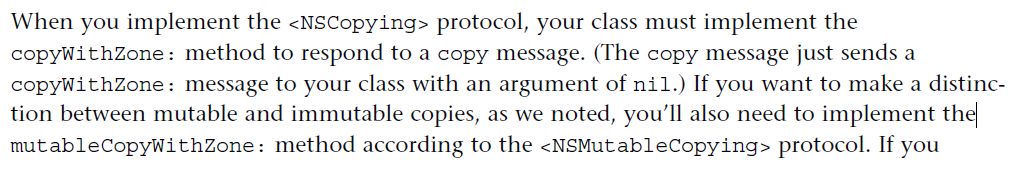


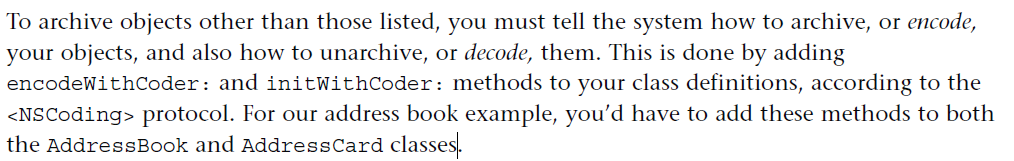






copies, by default, are *shallow* copies.





65、实例变量的访问域

@private：只能在本类中访问。

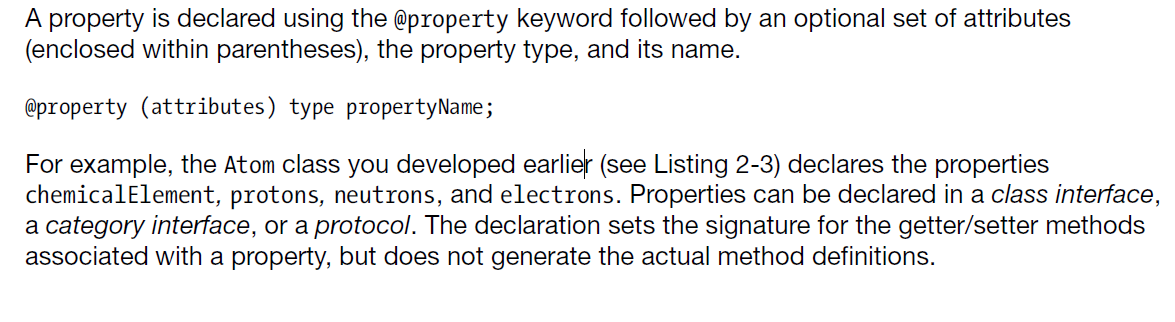
@protected：在本类和子类实例方法中访问，这是实例变量默认的访问域。

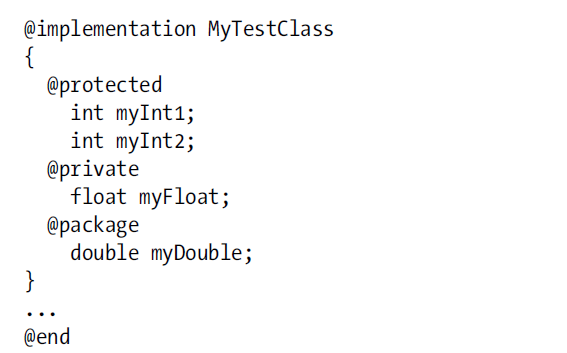
@public：可以在任何地方访问。

@package：在本包被认为是public的，在包外，被认为是私有的。

它们起作用只在interface中声明，在implementation中相当于private，即使是public，在子类中也不可见。可以通过->访问实例变量。

为了封装，通常声明properties，properties根据修饰符自动生成了get和set方法。



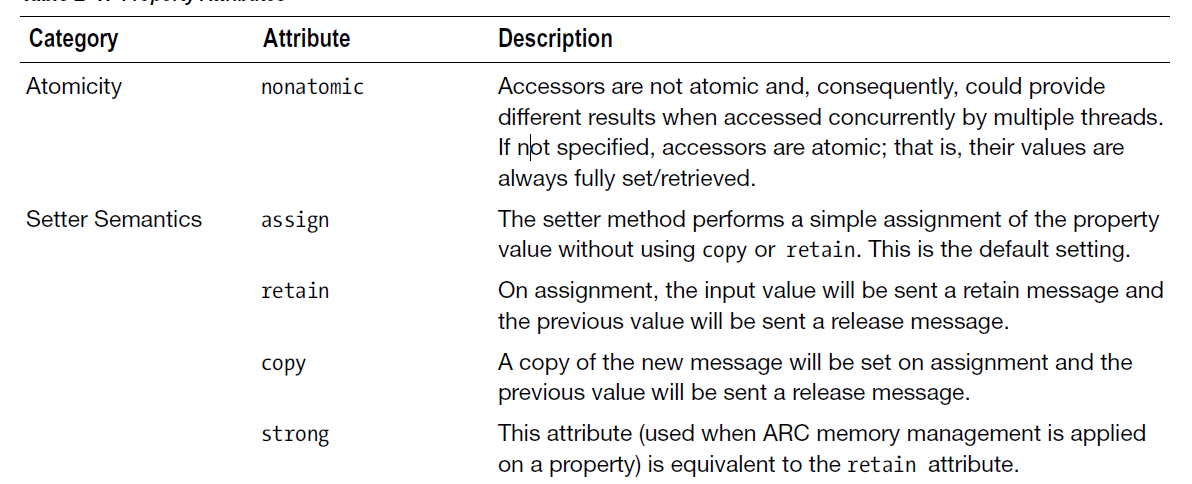


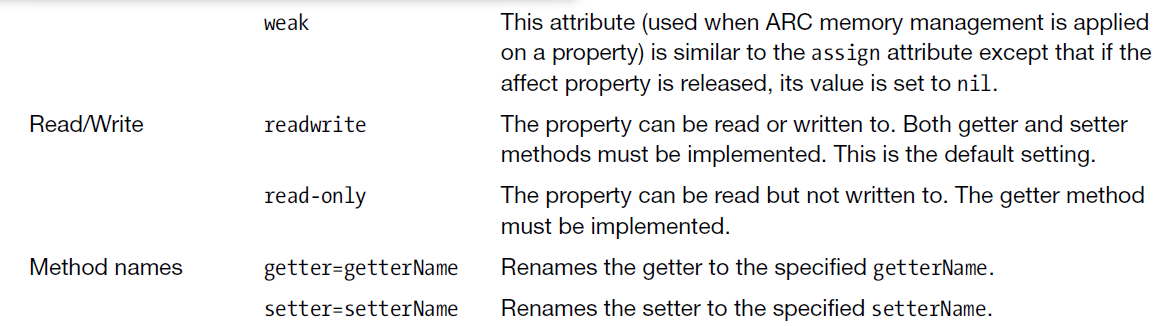
66、Properties声明方式

@property (attributes) type propertyName;//这是声明，不是定义，属性定义有三种：显式定义get，set（属性实例以\_开头）;systhesis @synthesize propertyName [= instanceVariableName]; 实例名可以忽略，编译器自动生成\_propertyName实例变量名，自动生成getset; Autosynthesis apple推荐的方式。自动systhesize属性。

(attributes)有如下几种

|  |  |
| --- | --- |
| nonatomic | 多线程访问时非原子的，如果不指定，这访问是原子的。指定后，因为是非原子的，多线程访问可能获取到不同的结果。 |
| assign | 这是默认的设定，简单的属性赋值，不会增加引用计数 |
| retain | On assignment,the input value will be sent a retain message and the previous value will be sent a release message.retain内存计数加1 |
| copy | A copy message will be sent on assignment and the previous value will be sent a release message. copy的本质为复制该内存所存储的内容，重新创建一个对象赋给其相同的内容，很明显，在copy这个过程中也发生了一次retain，不过这是个全新的对象。   retain一般适用于OC中的对象   assign一般适用于非OC对象，如int等普通类型   copy一般适用于NSString等不可变的对象，因为是重新创建了对象，并且内容不变，因此不用担心后面的操作会对该属性的值产生影响。 |
| strong | 当ARC内存管理应用时，等价于retain |
| weak | 当ARC应用时，类似于assign,如果对象被释放时，the value is auto set to nil. |
| readwrite | Property可以读和写，这是默认设置 |
| read-only | 只能读， |
| getter=getterName | 重命名getter方法到getterName |
|  |  |
| setter=setterName | 重命名setter方法到setterName |





67、property访问有两种方式：方法（set加首字母大写实例名， 实例名）

OC提供了第二种. Dot notation方式。 Dot natation实际上调用了get和set方法。

myObject.propertyName;

myObject.propertyName = propertyValue;

67、A *protocol* declares methods and properties that can be implemented by any class.

@protocol ProtocolName

// Property declarations

@required

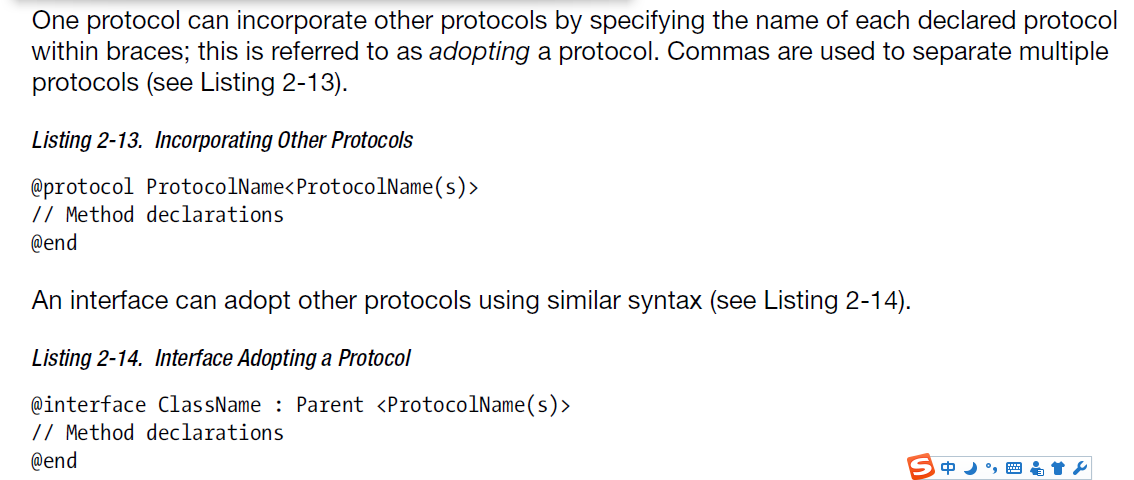
// Method declarations

@optional

// Method declarations

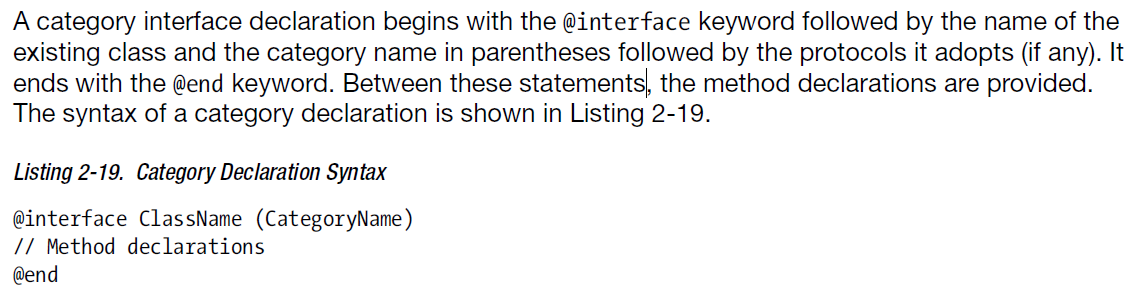
@end

Protocols can have both *required* and *optional* methods; optional methods do not require that an implementation of the protocol implement these methods. The directives @required and @optional (followed by the method name(s)) are used to mark a method appropriately. If neither keyword is specified, the default behavior is required.



68、Category

Category类似于C#中的拓展方法，enables the addition of new functionality to an existing class without subclassing it.



@implementation ClassName (CategoryName)

//method implementation

@end

69、Extensions

Extensions是匿名的category。在Extensions中声明的方法必须在main @implementation块中实现。

@interface ClassName ()

{

// Instance variable declarations

}

// Property declarations

// Method declarations

@end

an extension differs from a category in that it can declare instance variables and properties. The compiler will verify that the methods (and properties) declared in an extension are implemented. A class extension is commonly placed in the same file as the class implementation file and it is used to group and declare additional required, private methods (e.g., not part of the publicly declared API) for use solely within a class.

70、

# Ios

1. IOS图片应该采用.png格式。
2. Controller和objects in a storyboard交互使用叫做outlet的property。 By decalring an outlet and connecting that outlet to the ui object,you can use outlet to operate ui. The ui in storyboard can be set up to trigger special methods in our controller class, these methods as action methods.
3. Outlets are special objective-c properties that are declared using the keyword IBoutlet。

@property (weak, nonatomic) IBOutlet UIButton \*myButton;

Action are methods declared with return type IBAction，

- (IBAction)doSomething:(id)sender;

IBAction和void一样不需要返回值。

1. Every iOS application has exactly one instance of UIApplication。AppDelegate可以代理应用程序的行为。
2. Interface controls are used in three basic modes: active, static and passive. Active🡨🡪Button static 🡨 🡪label passive🡨🡪text
3. 使用Asset Catalog来管理image. RTImageAssets来辅助管理图片
4. 所有的子类of UIView都有一个NSInteger的tag。
5. #pargma mark –的“-”后面不能有空格，是用来做代码分割。
6. Auto Layout

<http://www.cocoachina.com/ios/20141020/9978.html>

<http://www.cocoachina.com/ios/20150421/11632.html>

# ARC及内存管理

参考文档：

<https://developer.apple.com/library/ios/releasenotes/ObjectiveC/RN-TransitioningToARC/Introduction/Introduction.html>

1. ARC在编译时自动添加retain,release and autorelease 到你的代码中。Compiler自动生成合适的dealloc。
2. Object properties are strong by default。