

随着 iOS 平台开发的职位的增加，笔试、面试也越来越有“套路”，这里我总结了一些面试题，多数是 Objective-C 的基础知识，适合于面试新人，答案是我自己答的，不准确的地方，欢迎指出。

1. Object-c 的类可以多重继承么？可以实现多个接口么？Category 是什么？重写一个类的方式用继承好还是分类好？为什么？

Object-c 的类不可以多重继承；可以实现多个接口，通过实现多个接口可以完成 C++ 的多重继承；Category 是类别，一般情况用分类好，用 Category 去重写类的方法，仅对本 Category 有效，不会影响到其他类与原有类的关系。

2. #import 跟 #include 又什么区别，@class 呢，#import<> 跟 #import” 又什么区别？

#import 是 Objective-C 导入头文件的关键字，#include 是 C/C++ 导入头文件的关键字，使用 #import 头文件会自动只导入一次，不会重复导入，相当于 #include 和 #pragma once；@class 告诉编译器某个类的声明，当执行时，才去查看类的实现文件，可以解决头文件的相互包含；#import<> 用来包含系统的头文件，#import” 用来包含用户头文件。www.syzw168.com
www.ccbz.com

3. 属性 readonly, readonly, assign, retain, copy, nonatomic 各是什么作用，在那种情况下用？

readonly 是可读可写特性；需要生成 getter 方法和 setter 方法时

readonly 是只读特性 只会生成 getter 方法 不会生成 setter 方法 ;不希望属性在类外改变

assign 是赋值特性，setter 方法将传入参数赋值给实例变量；仅设置变量时；

retain 表示持有特性，setter 方法将传入参数先保留，再赋值，传入参数的 retaincount 会+1;

copy 表示赋值特性，setter 方法将传入对象复制一份；需要完全一份新的变量时。

nonatomic 非原子操作，决定编译器生成的 setter getter 是否是原子操作，atomic 表示多线程安全，一般使用 nonatomic

4. 写一个 setter 方法用于完成@property (nonatomic, retain) NSString *name, 写一个 setter 方法用于完成@property (nonatomic, copy) NSString *name

```
[cpp] view plaincopy
```

```
01.- (void) setName:(NSString*) str
```

```
02.{
03.    [str retain];
04.    [name release];
05.    name = str;
06.}
07.- (void)setName:(NSString *)str
08.{
09.    id t = [str copy];
10.    [name release];
11.    name = t;
12.}
```

5.对于语句 `NSString*obj = [[NSData alloc] init];` obj 在编译时和运行时分别是什么类型的对象?

编译时是 `NSString` 的类型；运行时是 `NSData` 类型的对象

6.常见的 `object-c` 的数据类型有那些，和 `C` 的基本数据类型有什么区别？如：`NSInteger` 和 `int`

`object-c` 的数据类型有 `NSString`, `NSNumber`, `NSArray`, `NSMutableArray`, `NSData` 等等，这些都是 `class`，创建后便是对象，而 `C` 语言的基本数据类型 `int`，只是一定字节的内存空间，用于存放数值；而 `object-c` 的 `NSNumber` 包含有父类 `NSObject` 的方法和 `NSNumber` 自己的方法，可以完成复杂的操作。

7.id 声明的对象有什么特性？

id 声明的对象具有运行时的特性，即可以指向任意类型的 `objective-c` 的对象；

8.Objective-C 如何对内存管理的,说说你的看法和解决方法？

Objective-C 的内存管理主要有三种方式 `ARC`（自动内存计数）、手动内存计数、内存池。

9.内存管理的几条原则是什么？按照默认法则.那些关键字生成的对象

需要手动释放？在和 `property` 结合的时候怎样有效的避免内存泄露？

谁申请，谁释放

遵循 Cocoa Touch 的使用原则:

内存管理主要要避免“过早释放”和“内存泄漏”，对于“过早释放”需要注意@property 设置特性时，一定要用对特性关键字，对于“内存泄漏”，一定要申请了要负责释放，要细心。

关键字 alloc 或 new 生成的对象需要手动释放;

设置正确的 property 属性，对于 retain 需要在合适的地方释放，

10.如何对 iOS 设备进行性能测试?

Profile-> Instruments ->Time Profiler

11.看下面的程序,第一个 NSLog 会输出什么? 这时 str 的 retainCount 是多少? 第二个和第三个呢? 为什么?

```
[cpp] view plaincopy
01.=====
02.NSMutableArray* ary = [[NSMutableArray array] retain];
03.NSString *str = [NSString stringWithFormat:@"%test"];
04.[strretain];
05.[aryaddObject:str];
06.NSLog(@"%@%d",str,[str retainCount]);
07.[strretain];
08.[strrelease];
09.[strrelease];
10.NSLog(@"%@%d",str,[str retainCount]);
11.[aryremoveAllObjects];
12.NSLog(@"%@%d",str,[str retainCount]);
13.=====
```

str 的 retainCount 创建+1， retain+1， 加入数组自动+1

3

retain+1, release-1, release-1

2

数组删除所有对象， 所有数组内的对象自动-1

1

12. Object C 中创建线程的方法是什么？如果在主线程中执行代码，方法是什么？如果想延时执行代码、方法又是什么？

线程创建有三种方法：使用 NSThread 创建、使用 GCD 的 dispatch、使用子类化的 NSOperation，然后将其加入 NSOperationQueue；在主线程执行代码，方法是 performSelectorOnMainThread，如果想延时执行代码可以用 performSelector:onThread:withObject:waitUntilDone:

13.描述一下 iOS SDK 中如何实现 MVC 的开发模式

MVC 是模型、视图、控制开发模式，对于 iOS SDK，所有的 View 都是视图层的，它应该独立于模型层，由视图控制层来控制。所有的用户数据都是模型层，它应该独立于视图。所有的 ViewController 都是控制层，由它负责控制视图，访问模型数据。