

iOS经典面试题

1小时前

内牛满面ing 阅读：1426 评论：10 赞：9

ios

面试题

前言

写这篇文章的目的是因为前两天同学想应聘iOS开发，从网上找了iOS面试题和答案让我帮忙看看。我扫了一眼，倒吸了一口冷气，仔细一看，气的发抖。整篇题目30多个没有一个答案是对的，总结这篇面试题的作者对iOS机制根本就是一知半解就敢发到网上，不管有心还是无心都是坑害新人。所以在这里总结一下这几年面试别人和被别人面试遇到的一些我认为比较好的基础题目分享给大家，进阶题目在后续补充。我的理解如果有错漏请一定指出，非常感谢！

从12年开始做面试官到现在已经三个年头了，这期面过很多形形色色的开发，有腾讯的、有百度的、有小公司的、有做外包的，还有完全没有代码能力来做管理的。给我的感觉是百度出品的技术能力最好，基础知识、技术细节掌握的非常的扎实并且自信。腾讯出品技术细节上也非常厉害，平均来说比百度差一些，但是解决问题的能力非常强，思维很活跃大局观好。小公司、外包公司的同学技术细节、基础知识要差一些。我个人是非常反对以出身论英雄BAT出品就一定比其他公司的优秀，只能说BAT提供的技术氛围更好，你可以跟更多优秀的人共事，相比小公司技术进步更容易些。

这期间我也面了很多互联网公司，腾讯、百度、阿里、快播还有其他很多中小型公司，给我的感觉是中国整体的互联网面试官的态度非常糟糕，面试官经常性习惯性的迟到，理由有开会啦、在吃饭啦，最离谱的是没有任何理由就让你在那里等。其实我觉得现在这些公司啊还没有看清楚一个现状，就是绝大多数干开发的都只是来看看机会，不是来求职的，所以总是找不到人。一边说自己的公司多么求贤若渴，应聘者来了又不给予应有的尊重。我认为面试是非常不好的一个词，应聘过程应该叫面谈、review更合适，它是一个双向选择的过程。

我参加的面试不下五十次，只有两次面试官递给我一杯水，一次是腾讯的技术一面，后来那个面试官也是我最尊敬的组长，还有一次是快播的hr面。其他面试官直接无视汗流浹背顶着深圳烈日赶来面试的我，基本上每次面试完都是半脱水状态。面试更主要的是考察一个应聘者在正常环境下的工作能力而不是极端环境下的应激能力，面试官不是要把应聘者问倒、提一些刁钻的问题把他难住而是应该帮助应聘者缓解他的紧张和压力，并适当的提示，让应聘者最大限度的发挥，这才是一个合格面试官最基本的素质。

iOS面试知识点

现在进入本篇的正题。本篇的面试题是我认为比较好的iOS开发基础知识点，希望大家看过这后在理解的基础上掌握而不是死记硬背。死记硬背很快也会忘记的。

1 iOS基础

1.1 父类实现深拷贝时，子类如何实现深度拷贝。父类没有实现深拷贝时，子类如何实现深度拷贝。深拷贝同浅拷贝的区别：浅拷贝是指针拷贝，对一个对象进行浅拷贝，相当于对指向对象的指针进行复制，产生一个新的指向这个对象的指针，那么就是有两个指针指向同一个对象，这个对象销毁后两个指针都应该置空。深拷贝是对一个对象进行拷贝，相当于对对象进行复制，产生一个新的对象，那么就有两个指针分别指向两个对象。当一个对象改变或者被销毁后拷贝出来的新的对象不受影响。

实现深拷贝需要实现NSCoying协议，实现- (id)copyWithZone:(NSZone *)zone 方法。当对一个property属性含有copy修饰符的时候，在进行赋值操作的时候实际上就是调用这个方法。

父类实现深拷贝之后，子类只要重写copyWithZone方法，在方法内部调用父类的copyWithZone方法，之后实现自己的属性的处理

父类没有实现深拷贝，子类除了需要对自己的属性进行处理，还要对父类的属性进行处理。

1.2 KVO，NSNotification，delegate及block区别

KVO就是cocoa框架实现的观察者模式，一般同KVC搭配使用，通过KVO可以监测一个值的变化，比如View的高度变化。是一对多的关系，一个值的变化会通知所有的观察者。NSNotification是通知，也是一对多的使用场景。在某些情况下，KVO和NSNotification是一样的，都是状态变化之后

写博客 (/writer)



(http://blog.cocoachina.com/547930/1)

内牛满面ing

(http://blog.cocoachina.com/547930/1)

阅读：1525

评论：11

赞：10

告知对方。NSNotification的特点，就是需要被观察者先主动发出通知，然后观察者注册监听后再来进行响应，比KVO多了发送通知的一步，但是其优点是监听不局限于属性的变化，还可以对多种多样的状态变化进行监听，监听范围广，使用也更灵活。

delegate 是代理，就是我不想做的事情交给别人做。比如狗需要吃饭，就通过delegate通知主人，主人就会给他做饭、盛饭、倒水，这些操作，这些狗都不需要关心，只需要调用delegate（代理人）就可以了，由其他类完成所需要的操作。所以delegate是一对一关系。

block是delegate的另一种形式，是函数式编程的一种形式。使用场景跟delegate一样，相比delegate更灵活，而且代理的实现更直观。

KVO一般的使用场景是数据，需求是数据变化，比如股票价格变化，我们一般使用KVO（观察者模式）。delegate一般的使用场景是行为，需求是需要别人帮我做一件事情，比如买卖股票，我们一般使用delegate。Notification一般是进行全局通知，比如利好消息一出，通知大家去买入。delegate是强关联，就是委托和代理双方互相知道，你委托别人买股票你就需要知道经纪人，经纪人也不要知道自己的顾客。Notification是弱关联，利好消息发出，你不需要知道是谁发的也可以做出相应的反应，同理发消息的人也不需要知道接收的人也可以正常发出消息。

1.3 KVC如果实现，如何进行键值查找。KVO如何实现 请看这两篇博文 KVC KVO

1.4 将一个函数在主线程执行的4种方法 GCD方法，通过向主线程队列发送一个block块，使block里的方法可以在主线程中执行。 dispatch_async(dispatch_get_main_queue(), ^{
//需要执行的方法 }); NSOperation 方法 NSOperationQueue mainQueue = [NSOperationQueue mainQueue]; //主队列 NSBlockOperation operation = [NSBlockOperation blockOperationWithBlock:^(//需要执行的方法);
[mainQueue addOperation:operation]; NSThread 方法 [self performSelector:@selector(method) onThread:
[NSThread mainThread] withObject:nil waitUntilDone:YES modes:nil];

[self performSelectorOnMainThread:@selector(method) withObject:nil waitUntilDone:YES];
[[NSThread mainThread] performSelector:@selector(method) withObject:nil]; RunLoop方法 [[NSRunLoop mainRunLoop] performSelector:@selector(method) withObject:nil]; 1.5 如何让计时器调用一个类方法 计时器只能调用实例方法，但是可以在这个实例方法里面调用静态方法。使用计时器需要注意，计时器一定要加入RunLoop中，并且选好model才能运行。scheduledTimerWithTimeInterval方法创建一个计时器并加入到RunLoop中所以可以直接使用。如果计时器的repeats选择YES说明这个计时器会重复执行，一定要在合适的时机调用计时器的invalid。不能在dealloc中调用，因为一旦设置为repeats 为yes，计时器会强持有self，导致dealloc永远不会被调用，这个类就永远无法被释放。比如可以在viewDidDisappear中调用，这样当类需要被回收的时候就可以正常进入dealloc中了。
[NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(timerMethod) userInfo:nil repeats:YES];

-(void)timerMethod { //调用类方法 [[self class] staticMethod]; }
-(void)invalid { [timer invalid]; timer = nil; } 1.6 如何重写类方法 1、在子类中实现一个同基类名字一样的静态方法 2、在调用的时候不要使用类名调用，而是使用[self class]的方式调用。原理，用类名调用是早绑定，在编译期绑定，用[self class]是晚绑定，在运行时决定调用哪个方法。 1.7 NSTimer创建后，会在哪个线程运行。用 scheduledTimerWithTimeInterval创建的，在哪个线程创建就会被加入哪个线程的RunLoop中就运行在哪个线程 自己创建的Timer，加入到哪个线程的RunLoop中就运行在哪个线程。 1.8 id和NSObject*的区别 id是一个 objc_object 结构体指针，定义是 typedef struct objc_object *id id可以理解为指向对象的指针。所有oc的对象 id都可以指向，编译器不会做类型检查，id调用任何存在的方法都不会在编译阶段报错，当然如果这个id指向的对象没有这个方法，该崩溃还是会崩溃的。

NSObject *指向的必须是NSObject的子类，调用的也只能是NSObjec里面的方法否则就要做强制类型转换。
不是所有的OC对象都是NSObject的子类，还有一些继承自NSProxy。NSObject *可指向的类型是id的子集。

我的理解如果有错漏请一定指出，非常感谢！ 以下内容后续补充 iOS 核心框架 CoreAnimation CoreGraphics CoreLocation AVFoundation Foundation iOS核心机制 UITableView 重用 ObjC内存管理；自动释放池，ARC如何实现 runloop runtime Block的定义、特性、内存区域、如何实现 Responder Chain NSOperation GCD 数据结构 8大排序算法 二叉树实现 二分查找实现 面向对象编程 封装、继承、多态

设计模式6个原则
设计一个类的功能，如何划分粒度（单一职责）
接口隔离。
如果有一个鸟类，有飞的动作，一个鸵鸟继承它是合适的吗（里氏替换）
类之间的依赖如何依赖耦合度最小（依赖倒转） 高层依赖低层，低层不能依赖高层。依赖接口，不能依赖具体的类。

如果A要调用C函数， 但C是B的成员类，应该如何设计？（迪米特）

如何设计类， 能做到只增加代码， 而不修改代码， 有哪些经验（开放封闭） 通过设计模式解决。

计算机技术 计算机网络： TCP/IP、HTTPCDN、SPDY 计算机安全： RSA、AES、DES 操作系统： 线程、进程、堆栈、死锁、调度算法 iOS新特性、新技术 iOS7 UIDynamic、SpritKit、新布局、扁平化 iOS8 应用程序扩展、HealthKit、SceneKit、CoreLocation、TouchID、PhotoKit iOS9 Apple Watch 第三方库：SDWebImage、AFNetwork、JSONKit、wax swift

收藏

分享到:   

相关博文

OS X 强制退出 (http://blog.cocoachina.com/article/13339)	ZYThumbnailTableView---堪比一个小型阅读App (http://blog...
无标题文章 (http://blog.cocoachina.com/article/13535)	波浪进度指示器吧WaveLoadingIndicator (http://blog.cocoac...
SpreadButton,想拥有一个应用内的AssistiveTouch按钮吗 (htt...	提供一种比较简单的单选表的实现 (http://blog.cocoachina.co...
libxml/tree.h file not found 的解决办法 (http://blog.cocoachin...	卡拉OK效果的实现-IOS音乐播放器 (http://blog.cocoachina.c...
第三方开源库 (http://blog.cocoachina.com/article/12350)	navigation 栏的一些设置 (http://blog.cocoachina.com/article/...

我来说两句



你怎么看？ 快来评论一下吧！

您还没有登录！ 请 [登录 \(/signin\)](/signin) 或 [注册 \(/signup\)](/signup)

发表评论

所有评论（10）



yuexiaoxing (http://www.cocoachina.com/bbs/u.php?action=feed&uid=286271)2016-03-10 17:23:24

好多赞， 虽然n久不搞这个了， 不过还是喜欢oc的设计理念， 抽空来看下，
(http://www.cocoachina.com/bbs/u.php?action=feed&uid=286271)

回复



xiaowu0125 (http://www.cocoachina.com/bbs/u.php?action=feed&uid=529497)2016-03-10 16:11:18

感谢楼主 很有用
(http://www.cocoachina.com/bbs/u.php?action=feed&uid=529497)

回复



漓清i (http://www.cocoachina.com/bbs/u.php?action=feed&uid=499847)2016-03-09 15:19:24

Copying协议楼主写错了 Coying
(http://www.cocoachina.com/bbs/u.php?action=feed&uid=499847)

回复



liandwj (http://www.cocoachina.com/bbs/u.php?action=feed&uid=426834)2016-03-08 14:10:02

"...面试官经常性习惯性的迟到， 理由有开会啦、在吃饭啦， 最离谱的是没有任何理由就让你在那里等。"， 非常同意， 上次去面试遇到过领导不在， 去总部开会了， 让我改天再来的情况。。。
(http://www.cocoachina.com/bbs/u.php?action=feed&uid=426834)

回复



Terroy (http://www.cocoachina.com/bbs/u.php?action=feed&uid=500235)2016-03-08 13:33:08

楼主快出后续的文章吧， 写的太好了， 只是该好好排版啊
(http://www.cocoachina.com/bbs/u.php?action=feed&uid=500235)

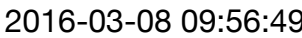
回复



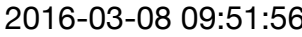
mr.nie (http://www.cocoachina.com/bbs/u.php?action=feed&uid=364187)2016-03-08 11:32:27

写好文章得耐住性子， 不过总结的都是经验

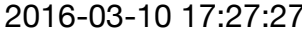
回复



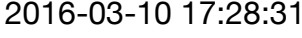
回复



回复



回复



回复

京网文[2012]0426-138号 (<http://182.131.21.137/ccnt-apply/admin/business/preview/business-446CE9DA949026D63939F>)