OC技术面试参考：<https://www.jianshu.com/p/e709fde38de3>

1、说说你认识的Swift是什么？

Swift是苹果于2014年WWDC（苹果开发者大会）发布的新开发语言，可与Objective-C共同运行于MAC OS和iOS平台，用于搭建基于苹果平台的应用程序。

2、举例说明Swift里面有哪些是 Objective-C中没有的？

Swift引入了在Objective-C中没有的一些高级数据类型，例如tuples（元组），可以使你创建和传递一组数值。

wift还引入了可选项类型（Optionals），用于处理变量值不存在的情况。可选项的意思有两种：一是变量是存在的，

例如等于X，二是变量值根本不存在。Optionals类似于Objective-C中指向nil的指针，但是适用于所有的数据类型，而非仅仅局限于类，Optionals 相比于Objective-C中nil指针更加安全和简明，并且也是Swift诸多最强大功能的核心。

3、NSArray与NSSet的区别？

NSArray内存中存储地址连续，而NSSet不连续

NSSet效率高，内部使用hash查找；NSArray查找需要遍历

NSSet通过anyObject访问元素，NSArray通过下标访问

4、Swift比Objective-C有什么优势？

Swift全面优于Objective-C语言，性能是Objective-C的1.3倍，上手更加容易。

5、NSHashTable与NSMapTable？

NSHashTable是NSSet的通用版本，对元素弱引用，可变类型；可以在访问成员时copy

NSMapTable是NSDictionary的通用版本，对元素弱引用，可变类型；可以在访问成员时copy

(注：NSHashTable与NSSet的区别：NSHashTable可以通过option设置元素弱引用/copyin，只有可变类 型。但是添加对象的时候NSHashTable耗费时间是NSSet的两倍。

NSMapTable与NSDictionary的区别：同上)

6、Swift 是一门安全语言吗？

Swift是一门类型安全的语言，Optionals就是代表。Swift能帮助你在类型安全的环境下工作，如果你的代码中需要使用String类型，Swift的安全机制能阻止你错误的将Int值传递过来，这使你在开发阶段就能及时发现并修正问题。

7、属性关键字assign、retain、weak、copy

assign：用于基本数据类型和结构体。如果修饰对象的话，当销毁时，属性值不会自动置nil，可能造成野指针。

weak：对象引用计数为0时，属性值也会自动置nil

retain：强引用类型，ARC下相当于strong，但block不能用retain修饰，因为等同于assign不安全。

strong：强引用类型，修饰block时相当于copy。

8、weak属性如何自动置nil的？

Runtime会对weak属性进行内存布局，构建hash表：以weak属性对象内存地址为key，weak属性值(weak自身地址)为value。当对象引用计数为0 dealloc时，会将weak属性值自动置nil。

9、Swift 是一门安全语言吗？

Swift是一门类型安全的语言，Optionals就是代表。Swift能帮助你在类型安全的环境下工作，如果你的代码中需要使用String类型，Swift的安全机制能阻止你错误的将Int值传递过来，这使你在开发阶段就能及时发现并修正问题。

 10、内存泄露问题？

主要集中在循环引用问题中，如block、NSTime、perform selector引用计数问题。

11、Block的循环引用、内部修改外部变量、三种block

block强引用self，self强引用block

内部修改外部变量：block不允许修改外部变量的值，这里的外部变量指的是栈中指针的内存地址。

\_\_block的作用是只要观察到变量被block使用，就将外部变量在栈中的内存地址放到堆中。

三种block：

NSGlobalBlack(全局)、

NSStackBlock(栈block)、

NSMallocBlock(堆block)

12、KVO底层实现原理？手动触发KVO？swift如何实现KVO？

KVO原理：当观察一个对象时，runtime会动态创建继承自该对象的类，并重写被观察对象的setter方法，重写的setter方法会负责在调用原setter方法前后通知所有观察对象值得更改，最后会把该对象的isa指针指向这个创建的子类，对象就变成子类的实例。

如何手动触发KVO：在setter方法里，手动实现NSObject两个方法：willChangeValueForKey、didChangeValueForKey

swift的kvo：继承自NSObject的类，或者直接willset/didset实现。

13、categroy为什么不能添加属性？怎么实现添加？与Extension的区别？category覆盖原类方法？多个category调用顺序

Runtime初始化时categroy的内存布局已经确定，没有ivar，所以默认不能添加属性。

使用runtime的关联对象，并重写setter和getter方法。

Extenstion编译期创建，可以添加成员变量ivar，一般用作隐藏类的信息。必须要有类的源码才可以添加，如NSString就不能创建Extension。

category方法会在runtime初始化的时候copy到原来前面，调用分类方法的时候直接返回，不再调用原类。如何保持原类也调用(https:*//www.jianshu.com/p/40e28c9f9da5)。*

多个category的调用顺序按照：Build Phases ->Complie Source 中的编译顺序。

14、load方法和initialize方法的异同。——主要说一下执行时间，各自用途，没实现子类的方法会不会调用父类的？

load initialize 调用时机 app启动后，runtime初始化的时候 第一个方法调用前调用 调用顺序 父类->本类->分类 父类->本类(如果有分类直接调用分类，本类不会调用) 没实现子类的方法会不会调用父类的 否 是 是否沿用父类实现 否 是

见图 1

15、对 runtime 的理解。——主要是方法调用时如何查找缓存，如何找到方法，找不到方法时怎么转发，对象的内存布局

OC中向对象发送消息时，runtime会根据对象的isa指针找到对象所属的类，然后在该类的方法列表和父类的方法列表中寻找方法执行。如果在最顶层父类中没找到方法执行，就会进行消息转发：Method resoution（实现方法）、fast forwarding（转发给其他对象）、normal forwarding（完整消息转发。可以转发给多个对象）

16、runtime 中，SEL和IMP的区别?

每个类对象都有一个方法列表，方法列表存储方法名、方法实现、参数类型，SEL是方法名(编号)，IMP指向方法实现的首地址

17、autoreleasepool的原理和使用场景?

若干个autoreleasepoolpage组成的双向链表的栈结构，objc\_autoreleasepoolpush、objc\_autoreleasepoolpop、objc\_autorelease

使用场景：多次创建临时变量导致内存上涨时，需要延迟释放

autoreleasepoolpage的内存结构：arm32是4k存储大小、arm64是16k存储大小

18、Autorelase对象什么时候释放？

在没有手加Autorelease Pool的情况下，Autorelease对象是在当前的runloop迭代结束时释放的，而它能够释放的原因是系统在每个runloop迭代中都加入了自动释放池Push和Pop。

19、Runloop与线程的关系？Runloop的mode? Runloop的作用？内部机制？

每一个线程都有一个runloop，主线程的runloop默认启动。

mode：主要用来指定事件在运行时循环的优先级

作用：保持程序的持续运行、随时处理各种事件、节省cpu资源(没事件休息释放资源)、渲染屏幕UI

20、iOS中使用的锁、死锁的发生与避免

@synchronized、信号量、NSLock等

死锁：多个线程同时访问同一资源，造成循环等待。GCD使用异步线程、并行队列

21、NSOperation和GCD的区别

GCD底层使用C语言编写高效、NSOperation是对GCD的面向对象的封装。对于特殊需求，如取消任务、设置任务优先级、任务状态监听，NSOperation使用起来更加方便。

NSOperation可以设置依赖关系，而GCD只能通过dispatch\_barrier\_async实现

NSOperation可以通过KVO观察当前operation执行状态(执行/取消)

NSOperation可以设置自身优先级(queuePriority)。GCD只能设置队列优先级 (DISPATCH\_QUEUE\_PRIORITY\_DEFAULT)，无法在执行的block中设置优先级

NSOperation可以自定义operation如NSInvationOperation/NSBlockOperation，而GCD执行任务可以自定义封装但没有那么高的代码复用度

GCD高效，NSOperation开销相对高

22、App启动优化策略？main函数执行前后怎么优化

启动时间 = pre-main耗时+main耗时

pre-main阶段优化：

删除无用代码

抽象重复代码

+load方法做的事情延迟到initialize中，或者+load的事情不宜花费太多时间

减少不必要的framework，或者优化已有framework

Main阶段优化

didFinishLauchingwithOptions里代码延后执行

首次启动渲染的页面优化

23、Swift 支持面向过程编程吗？

它采用了 Objective-C 的命名参数以及动态对象模型，可以无缝对接到现有的 Cocoa 框架，并且可以兼容 Objective-C 代码，支持面向过程编程和面向对象编程

24、Swift 是一门安全语言吗？

Swift是一门类型安全的语言，Optionals就是代表。Swift能帮助你在类型安全的环境下工作，如果你的代码中需要使用String类型，Swift的安全机制能阻止你错误的将Int值传递过来，这使你在开发阶段就能及时发现并修正问题。

25、Swift中如何定义变量和常量？

使用let来声明常量，使用var来声明变量

26、oc与js交互

拦截url

JavaScriptCore(只适用于UIWebView)

WKScriptMessageHandler(只适用于WKWebView)

WebViewJavaScriptBridge(第三方框架)

27、Swift的内存管理是怎样的？

Swift 使用自动引用计数（Automatic Reference Counting, ARC）来简化内存管理

28、struct、Class的区别

class可以继承，struct不可以

class是引用类型，struct是值类型

struct在function里修改property时需要mutating关键字修饰

29、访问控制关键字(public、open、private、filePrivate、internal)

public与open：public在module内部中，class和func都可以被访问/重载/继承，外部只能访问；而open都可以

private与filePrivate：private修饰class/func，表示只能在当前class源文件/func内部使用，外部不可以被继承和访问；而filePrivate表示只能在当前swift源文件内访问

internal：在整个模块或者app内都可以访问，默认访问级别，可写可不写

30、OC与Swift混编

OC调用swift：import "工程名-swift.h” @objc

swift调用oc：桥接文件

31、用Swift定义一个数组和字典？

let emptyArray = String[]()

let emptyDictionary = Dictionary<String, Float>()

32、try、try?与try!

try：手动捕捉异常

try?：系统帮我们处理，出现异常返回nil；没有异常返回对应的对象

try!：直接告诉系统，该方法没有异常。如果出现异常程序会crash

33、guard与defer

guard用于提前处理错误数据，else退出程序，提高代码可读性

defer延迟执行，回收资源。多个defer反序执行，嵌套defer先执行外层，后执行内层

34、架构&设计模式

MVC设计模式介绍

MVVM介绍、MVC与MVVM的区别？

ReactiveCocoa的热信号与冷信号

缓存架构设计LRU方案

SDWebImage源码，如何实现解码

AFNetWorking源码分析

组件化的实施，中间件的设计

哈希表的实现原理？如何解决冲突

35、数据结构&算法

快速排序、归并排序

二维数组查找(每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请完成一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数)

二叉树的遍历：判断二叉树的层数

单链表判断环

36、内存泄露问题？

主要集中在循环引用问题中，如block、NSTime、perform selector引用计数问题。

37、crash防护？

unrecognized selector crash

KVO crash

NSNotification crash

NSTimer crash

Container crash（数组越界，插nil等）

NSString crash （字符串操作的crash）

Bad Access crash （野指针）

UI not on Main Thread Crash (非主线程刷UI (机制待改善)