智游面试题

1、你如何理解OC这门语言的？你谈一下你对OC的理解？

OC语言是C语言的一个超集，只是在C的基础上加上了面向对象的语言特性，如：封装、继承、多 态。

OC语言优点：1. 支持类别　2. 可与c++混编

OC相比C++相比：1. 不支持命名空间　2. 不支持运算符重载　3. 不支持多继承

2、用过系统的哪些框架？

UIKit:　UI控件

Foundation:　基本的数据类型，NSString，NSArray等

CoreGraphics:　绘图相关的类和方法

CoreAnimation:　核心动画

1.音频和视频

Core Audio	OpenAL	Media Library	AV Foundation

2.数据管理

Core data	SQLite

3.图形和动画

Core Animation	OpenGL ES	Quartz 2D	Core Graphic

4.用户应用

Address Book	Core Location	Map Kit	Store Kit

3、C和OC如何混用，C++和OC如何混用？

实现文件的扩展名.m改成.mm即可，但cpp文件必须只能使用c/c++代码，而且cpp文件include的头文件中，也不能出现obj-c的代码，因为cpp只能写C++的代码。

4、最新版本的iOS有什么新特性？

iOS9中开发新特性:

1. 默认支持https
2. http需要在plist文件中单独配置，才能在程序中使用http

   在Info.plist中添加NSAppTransportSecurity类型Dictionary。

   在NSAppTransportSecurity下添加NSAllowsArbitraryLoads类型Boolean,值设为YES。
3. iPad中可以实现分屏功能。
4. 3D Touch

5. App Thinning， 可以给安装包瘦身

iOS8中开发新特性：

1. 定位功能需要在plist中单独配置
2. 推送需要授权
3. homekit、healthkit

注意：其他的可以看一下，因为适配的原因，一般不会在程序中使用最新的特性，例如如果你使用了iOS9的新特性，那么你发布的程序就要求用户的手机系统必须是iOS9及以上。

5、#include与#import的区别？#import与@class的区别?

1. **#include与#import的区别**：#include与#import其效果相同，只是后者不会引起重复导入，确保头文件只会被导入一次。

2. #import与@class的区别：import是导入头文件，会把头文件的所有信息获取到，这个类有哪些变量和方法。而@class只会告诉编译器，其后面声明的名称是类的名称，至于这些类是如何定义的，完全不知道，所以即使在头文件中使用了@class，还是需要在.m中导入头文件。

注意：使用@class 是为了防止头文件之间相互导入。

6、你平时是怎么描述NSString类型的？为什么用copy?

一般使用copy

假如一个对象的属性是NSString类型，而且使用了retain描述，如果把一个NSMutableString的对象赋值给这个属性（ 子类的对象可以赋给父类的指针），属性内部只是retain了一下，还是指向同一个nsmutablestring对象，如果将来在赋值完成之后，再次修改mutablestring的值，那么对象的属性的值也会发生改变，这样肯定是不对的，所以nsstring类型的属性一般使用copy，这样给对象的属性赋值完成之后，对象的属性是重新拷贝了一份，就不用害怕外部再修改，即使修改，也不会影响到属性的值。

深拷贝浅拷贝问题

浅拷贝拷贝的是指针，深拷贝拷贝的是对象。

7、assign、retain和copy之间的区别是什么？作用是什么?

assign：普通赋值，一般基于基本数据类型，常见委托设计模式，以此来防止循环引用。

retain：保留计数，获得到了对象的所有权。引用计数在原有基础上加1，一般对象类型都使用retain。

copy：用来复制对象，一般不可变对象都使用copy，例如NSString，还有解释全局的block变量。还有就是你需要复制一个对象的时候。

8、描述一下viewController的生命周期

当我们调用UIViewControllller的view时，

系统首先判断当前的 UIViewControllller是否存在view，如果存在直接返回view，

如果不存在的话，会调用loadview方法，

然后判断loadview方法是否是自定义方法，

如果是自定义方法，就执行自定义方法，

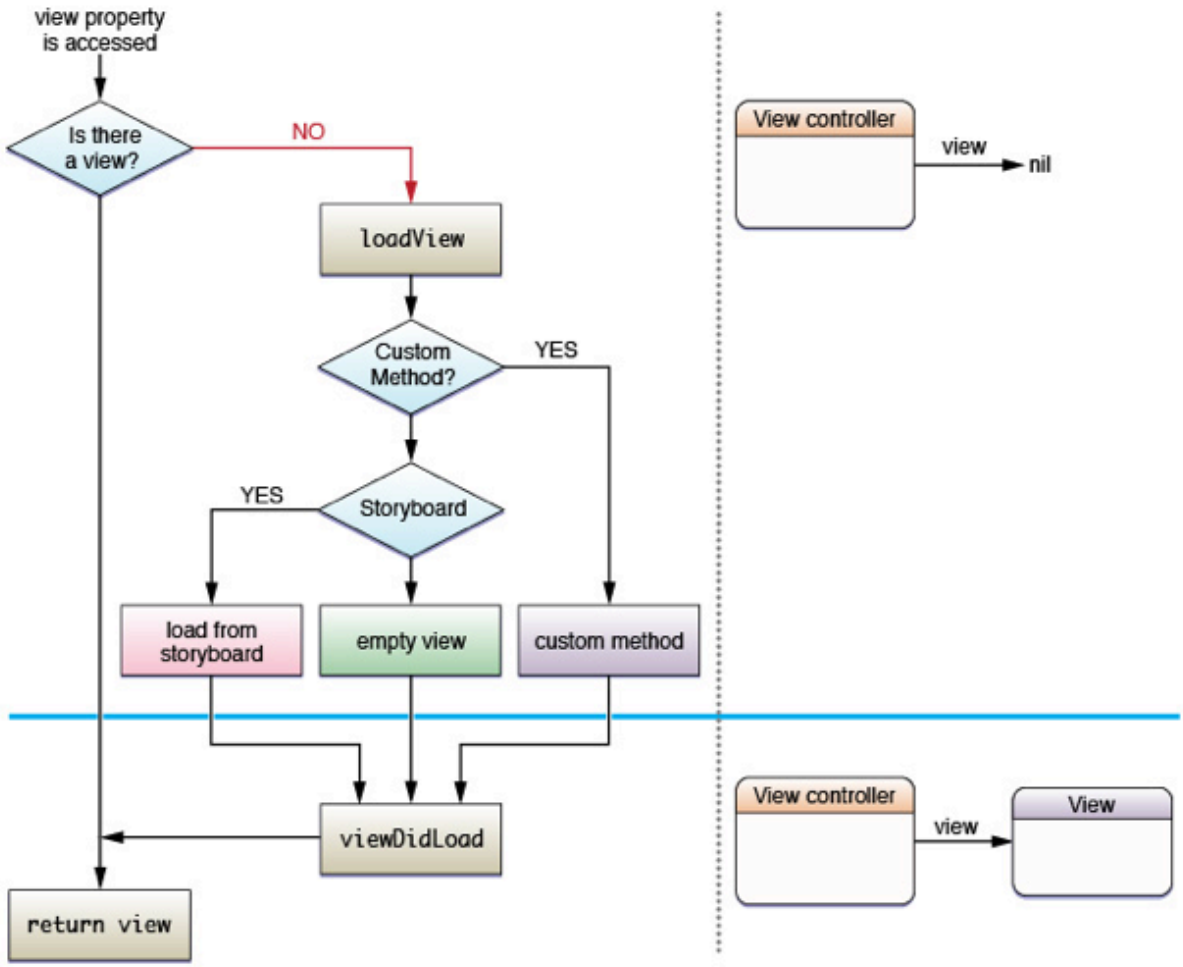
如果不是自定义方法，判断当时视图控制器是否有xib、stroyboard，

如果有xib、stroyboard加载xib、stroyboard，

如果没有创建一个空白的view，

调用viewDidLoad方法，

最后返回view。



9、你什么时候用自动释放池?

1. 创建一个分线程的时候，原来不会创建一个自动释放池，现在可以了，不用写也行。

2. 局部大量创建自动释放对象的时候，可以使用嵌套的自动释放池，及时释放局部自动释放对象，防止内存出现高峰。

10、你平时开发程序过程中是如何管理内存的?

参考回答：之前开发项目的时候普遍使用的是手动内存管理(MRC),现在都使用自动内存管理(ARC)来管理内存。

相关技术知识：

内存管理的本质是管理引用计数，但是我们在编写程序的时候，并不需要时刻关注引用计数，只需要按照一定的准则就可以管理好内存。

手动内存管理的关键就是对象引用计数所有权的问题，谁拥有这个对象引用计数的所有权，就要负责释放他所拥有的对象所有权，通过alloc，new，copy，retain这几个关键字我们可以获得对象的所有权，所以与之匹配的是，我们需要在合适的时候，通过release，autorelease等释放他所有拥有的对象所有权，不是我们拥有的对象引用计数，不需要我们负责释放，大家都各自按照这个规则来管理，才能确保引用计数的正确性。

ARC是自动的引用计数，是系统在编译阶段自动的帮助我们在代码中加入release，autorelease等内存管理代码，正因为是在编译阶段加入，所以并不会影响程序的执行效率，反而因为苹果的优化，相比手动内存管理，效率可能更高，但是使用ARC的时候，几个需要注意的问题就是：

1.局部变量被自动释放的问题。

2.ARC 中 OC对象和C结构体之间的桥接转换。

注意：

1. MRC、ARC中都会遇到的循环引用问题（两个strong指针相互引用，block中self指针的问题，代理使用assign）。

2. ARC和MRC可以混编，在ARC工程中，使用MRC，需要在bulid phase中的编译文件标志中加入 –fno-objc-arc ,表明这个文件使用MRC编译，反之，加入-fobjc-arc。

11、ARC中存在内存泄露吗?

ARC中如果内存管理不当的话，同样会存在内存泄露，例如：ARC中也会循环引用导致内存泄露，OC对象与CoreFoundation类之间桥接时，管理不当也会产生内存泄漏。

12、内存中堆和栈的区别

堆（heap），栈（stack）

alloc，new，copy的对象是创建在堆上的，需要自己负责管理，若程序员不释放，则内存溢出。
栈内存一般是系统自己创建并管理的，例如方法内的指针，形式参数等，系统会把这些变量放到栈中，并在方法结束的时候自动释放掉。

13、strong与weak的作用与区别

strong叫强引用，weak叫弱引用，强引用指向的对象不会被释放
一个对象没有强引用会立刻释放，弱引用指向的对象在释放时会自动置为空

14、block的内存问题？使用block需要注意什么？

1. 如果一个block被copy，block会对其中用到的对象retain一次，如果block内部用到了本类的属性和方法，也会把self retain一次，而block如果是全局的，这个类本身又copy了这个block，这个时候就形成了循环引用。

（大部分我们使用到block的情况都是这样的，如ASIHTTPRequest对象，设置asi.completionBlock:的时候，completionBlock作为asi对象的属性，肯定是被copy的，在传入的block内部，我们又会使用asi 对象，这就造成了asi对象copy了block，而block又retian了asi对象，造成循环引用，形成内存泄露）

2. 解决循环引用：将当前对象赋值给一个局部变量，并且使用__block关键字修饰该局部变量，使用该局部变量访问当前对象的属性和方法；arc中使用__weak代替。上面的例子中，我们应该__block ASIHTTPRequest *asi = …..； 这样创建asi对象。

15、内存不足，系统会发出警告，此时控制器应该如何处理？当内存不足的时候你改怎么处理？

内存不足时，系统会自动调用视图控制器的didReceiveMemoryWarning方法通知控制器内存不足，同时也会发送一个UIApplicationDidReceiveMemoryWarningNotification通知。
一般需要在此方法中，释放掉自己不需要使用的对象，如内存中缓存的图片数据等（想一想我们自己实现sdwebimage中的逻辑，就有关于这个方法的处理）。

16、你是如何理解表单元格重用机制 ？

当屏幕上的单元格滑出屏幕时，系统会把这个单元格添加到重用队列中，等待被重用，当有新单元格从屏幕外滑入屏幕内时，从重用队列中找看有没有可以重用的单元格，如果有，就拿过来用，如果没有就创建一个来使用。

17、iOS中数据存储有哪些方式？（数据持久化）每种存储方式各有什么特点？每种存储方式各自在什么场景下使用？

所有的本地持久化数据存储的本质都是写文件，而且只能存到沙盒中。
沙盒机制是苹果的一项安全机制，本质就是系统给每个应用分配了一个文件夹来存储数据，而且每个应用只能访问分配给自己的那个文件夹，其他应用的文件夹是不能访问的。

沙盒中有三个默认文件夹（支持自己创建新的）

Documents　存储用户相关的数据（用户拍的视频，用户创作的图片，用户唱的歌曲，用户收藏的商品）。

Library　跟程序相关的数据（程序缓存，程序配置文件）。

tmp　放临时文件，不需要永久存储的，比如下载的时候，需要存储到临时文件中，最终拷贝到Documents或Library中。

数据存储的核心都是写文件。主要有四种持久化方式：属性列表，对象序列化，SQLite 数据库 CoreData。

属性列表：应用于少量数据存储，比如登陆的用户信息，应用程序配置信息等。只有NSString，NSArray，NSDictory，NSData，可以WriteToFile；存储的依旧是plist文件，plist文件可以存储的7种数据类型：array，dictory，string，bool，data，date，number。

对象序列化：最终也是存为属性列表文件，如果程序中，需要存储的时候，直接存储对象比较方便，例如有一个设置类，我们可以把设置类的对象直接存储，就没必要再把里面的每一个属性单独存到文件中。对象序列化是将一个实现了NSCoding协议的对象，通过序列化（NSKeydArchiver）的形式，将对象中的属性抽取出来，转化成二进制流，也就是NSData，NSData可以选择write to file或者存储到NSUserdefault中。 必须实现的两个方法 encodeWithCoder，initWithCoder。对象序列化的本质就是 对象NSData。

SQLite：适合大量，重复，有规律的数据存储。而且频繁的读取，删除，过滤数据，这种适合使用数据库。

CoreData：Sqlite叫做关系型数据库，CoreData 是一中OR-Mapping的思想 ，O代表对象Object，R代表relationship，Mapping代表映射，直译过来就是对象关系映射，其实就是把对象的属性和表中的字段自动映射，简化程序员的负担，以面向对象的方式操作数据库。ORMapping是一种思想，CoreData实现了这种思想，在Java中，hibernate 也是对ORMapping的一种实现，只是利用java实现的。

CoreData本质还是数据库，只不过使用起来更加面向对象，不关注二维的表结构，而是只需要关注对象，纯面向对象的数据操作方式。我们直接使用数据库的时候，如果向数据库中插入数据，一般是把一个对象的属性和数据库中某个表的字段一一对应，然后把对象的属性存储到具体的表字段中.取一条数据的时候，把表中的一行数据取出，同样需要再封装到对象的属性中，这样的方式有点繁琐，不面向对象。CoreData解决的问题就是不需要这个中间的转换过程，看起来是直接把对象存储进去，并且取出来，不关心表的存在，实际内部帮你做好了映射关系。

CoreData中经常使用的类有哪些：

NSManagedObjectContext 管理对象上下文，相当于FMDB中的FMDatabase对象，我们对数据中的操作先存储到这个上下文中，然后把操作同步到数据库中。

NSManagedObject 托管对象。相当于是对表中一行数据的封装。

NSEntityDescription 实体描述。相当于在这个对象中定义了数据库中表的结构（比如包含哪些字段等）。

NSPersistentStoreCoordinator 持久化存储协调器。连接数据库的类，里面包含了，数据库的位置，名称等，相当于文件管理器，帮我们创建数据库文件等。

NSManagedObjectModel 托管对象模型。里面包含了数据库表，表之间关系的设计模型。其实这个对象里面包含的就是 我们使用coredata时，设计数据库模型 xcdatamodel文件中的信息。

18、如何优化tableView的使用?

1.复用单元格

2.单元格中的视图尽量都使用不透明的，单元格中尽量少使用动画

3.图片加载使用异步加载

4.滑动时不加载图片，停止滑动时开始加载

5.单元格中的内容可以在自定义cell类中的drawRect方法内自己绘制

6.如非必要，减少reloadData全部cell，只reloadRowsAtIndexPaths

7.如果cell是动态行高，计算出高度后缓存

8.cell高度固定的话直接使用cell.rowHeight设置高度

19、在做开发过程中是否使用过socket?

参考回答：了解过，但是实际项目中没有直接用过，了解过一些底层是socket实现的第三方sdk，如环信、爱萌等。

20、socket和http有什么区别?

socket 是网络传输层的一种技术，跟http有本质区别，http是应用层的一个网络协议。使用socket技术理论上来说，按照http的规范，完全可以使用socket来达到发送http请求的目的，只要发送的数据包按照http协议来即可。

socket与http的区别：

socket是长连接。http是短连接

socket是双向通信，http是单向的，只能客户端向服务器发送数据

socket 的数据完全由自己组织，http必须按照http协议来发送。

Socket使用场景：

1. 客户端频繁请求服务器，如股票应用， 需要一直向服务器请求最新的数据，如果使用http，那么第一：频繁请求，就会频繁连接，造成服务器压力巨大，如果用socket，一次连接，不会消耗服务器太多资源。第二：频繁发送，返回数据，如果使用http，因为http协议的限制，发送的数据包中包含了很多请求头，请求行等http协议必须带的数据，发送的数据量相比socket大很多，socket只需要请求和返回需要的数据即可，如股票应用中,只需要返回股票的最新价格即可，及时性会更高。

2. 客户端和服务器互相发送数据，如聊天应用，需要客户端上传聊天内容，同时，别人给你发消息，服务器也能主动把别人发送的消息发送给你。

需要使用socket技术的场景：网络游戏、即时通讯（一般不自己通过socket来实现，太过复杂，一般使用第三方平台，如环信，爱萌）、股票软件、自己实现推送机制。

21、什么是http？http协议的特点？http的数据包有哪几部分组成？GET请求和POST请求的区别？你还知道有哪些请求方式？Https是什么？

Http超文本传输协议，http是应用层的一个网络协议。

http特点：

1.短链接：是客户端主动发送请求，服务器做出响应，服务器响应之后，连接断开。

2.单向连接：服务器不能主动向客户端发送数据。

HTTP请求报文：

一个HTTP请求报文由请求行（request line），请求头（header），空行和请求数据4个部分组成。

请求行中规定了请求的方式（get/post）,请求的url，请求的http协议版本

请求头主要是传递一些参数，配置一些设置。常见的有range头，断点下载使用，cookie头，存储cookie信息，user-agent，表明客户端信息。

请求数据区放置post请求的数据。

HTTP响应报文：

HTTP响应由三个部分组成，分别是：状态行，响应头，响应正文。

状态行：组成：服务器协议版本、状态码，状态描述。常见的状态码有200（成功），404（url不存在），500（服务器内部错误）。一般以2开头的代表成功，以3开头的代表重定向，4开头的代表客户端错误，5开头的错误代表服务器端错误。

常用的请求方式是get，post。但是还有OPTIONS 、HEAD 、PUT、DELETE、TRACE，但是一般很少用。

GET请求：参数在地址后拼接，请求数据区没有请求数据，不安全（因为所有的参数都拼接在地址后面），不适合传输大量数据（长度有限制 ）

POST请求：参数在请求数据区放着，相对Get请求更安全，并且数据大小没有限制，一般上传文件使用。

无论是用get，post请求，如果url中含有特殊字符，如中文，特殊符号等，需要把url编码（字url字符串调用stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding）。

Https：
Https是：Https（Secure Hypertext Transfer Protocol）安全超文本传输协议，它是一个安全通信通道，它基于HTTP开发，用于在客户计算和服务器之间的交换信息。它使用安全套结字层（SSI）进行信息交换，简单来说它是HTTP的安全版。

22、你们项目的服务器是哪种语言开发的?

特别注意：服务器开发语言：我们公司用的是java（如果面试官问的话）。有些公司使用的是php、.net、asp、java、c、c++等。

23、在项目中你处理http通讯常用哪种第三方库?

在项目中，之前一般都使用ASIHTTPReqeust,因为ASIHTTPReqeust不再更新，不支持arc，所以现在都使用AFNetworking。目前在iOS9中发送http请求，需要在info.plist中开启HTTP请求。

ASIHTTPRequest，NSURLConnection是对CFNetwork的封装，相对更底层。AFNetworking是对NSURLConnection的封装。ASIHTTPRequest不支持ARC，不再更新，AFNetworking支持ARC，不断在更新。

AFNetworking默认支持的响应头格式比较少，不支持text/html，而一般服务器默认返回的类型就是text/html（想一想默认建立java web工程的时候，setContentType（text/html;charset = utf-8）），严格来说，如果服务器返回的数据是json格式，那么需要把响应头中的content-type 改为application/json,但是服务器程序员一般都不改，使用默认值，那么客户端如果使用AFNetworking，就会出现请求失败，所以一般使用AFNetworking，
manager.responseSerializer.acceptableContentTypes = [NSSet setWithObjects:@"text/html", nil];
需要修改一下这个属性，让他支持这个头。（如果面试官问，就说我们公司的情况，公司服务器返回的响应数据就是text/html的，所以需要设置）

24、你做项目的时候是如何区分手机网络类型的?

使用苹果官方的reachability 这个类，可以区分手机是否联网，并且能区分出是手机网络还是wifi网络，至于能不能 区分2g，3g，4g ，应该是不能，我项目中也没遇到这种情况。

25、TCP和UDP有什么不同?

tcp和udp都是网络传输层的协议，tcp提供可靠的数据连接，udp提供不可靠的数据连接，不会对数据包的顺序，是否丢失进行校验，如果丢失也不会重新发送，但是tcp会验证数据包的顺序，丢失还有重新发送，所以是可靠的。但是udp的优点正是因为少了这些校验，及时性更好一些，所以向常见的视频聊天，音频聊天都使用udp协议，即使丢一两个包，也无妨，最多声音模糊一下，或者画面稍微卡顿。

Tcp是面向连接的，一对一的通信，udp是广播方式，一对多的方式。我们使用socket的时候，可以选择使用tcp或者udp。

Tcp 有三次握手，udp没有，tcp连接的三次握手:

1.第一次握手: 客户端发送syn包（syn = j）到服务器，并进入SYN_SEND状态，等待服务器确认；

2.第二次握手：服务器收到syn包，必须确认客户的SYN（ack = j + 1），同时自己也发送一个SYN包（syn = k），即SYN + ACK包，此时服务器进入SYN + RECV状态；

3.第三次握手：客户端到服务器的SYN + ACK包，向服务器发送确认包ACK（ack = k + 1）,此时发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

完成三次握手，客户端与服务器开始传送数据。

26、网络七层协议，socket是属于那一层的，http是属于那一层的?

应用层、表示层、会话层、传输层、网络层、数据链路层、物理层

应用层

1.主要功能：用户接口，应用程序

2.application典型设备：网关

3.典型协议，标准和应用：TELNET FTP，HTTP

我们做 应用层，比如我们做软件，一个视频播放器，这个就是指一个应用层。

表示层

1.主要功能：数据的表示，压缩和加密presentation

2.典型设备：网关

3.典型协议，标准和应用：ASCLL PLCT TIFF JPEG MIDI MPEG

表示层相当于一个东西怎么表示，表示的一些协议，像图片：JEPG 声音：MDI 视频：MPEG

表示层就是定义这个层的协议的，比如：说某个人说说自己做表示层，可能这个人就是在做MPEG4

会话层

1.主要功能：会话的建立和结束的session

2.典型设备：网关

3.典型协议，标准 和应用：RPC SQL NFS X WINDOWS，ASP

传输层

1.主要功能：端到端控制transport

2.典型设备：网关

3.典型协议，标准和应用：TCP UDP SPX

网络层

1.主要功能：路由，寻址network

2.典型设备：路由器

3.典型协议，标准和应用：IP IPX APPETALK ICMP

数据链路层

1.主要功能：保证误差错的数据链路data link

2.典型设备：交换机，网桥，网卡

3.典型协议，标准和应用：802.2，802.3ATM,HDLC,FRAME RELAY

物理层

1.主要功能传输比特流physical

2.典型设备：集线器，中继器

3.典型协议，标准和应用：V.35,EIA/TIA-232

socket，tcp，udp属于传输层，http，ftp是属于应用层。不需要记住每一层的作用，只需要记住名称即可，而且需要知道，下层为上层提供服务。

27、断点续传是如何实现的?

所谓断点续传，也就是再次下载的时候，不是下载整个文件内容，而是从已经下载过的数据开始，下载剩下的所有数据 。

所以在客户端给服务器发送请求的时候，需要在请求头中加上range头，bytes=xx-xxx。xx代表需要下载文件的起始位置，xxx代表结束位置，xxx一般不填，代表从起始位置开始剩下的所有数据。

28、常用的数据组织格式有哪些？你平常是怎么样解析？有哪些数据解析方式？XML和JSON对比，它们有什么优缺点？

XML和JSON

平常一般我们公司服务器返回的数据格式是json，我一般通过系统的NSJSONSerialization来完成解析。之前系统没有这个类的时候，一般使用第三方库（JSONKit、SBJSON）。

XML有两种解析方式，Dom和Sax，一般我们都不使用XML这种格式，系统的NSXMLParser可以解析XML，属于Sax方式。

Dom方式是先读取文档的整个内容，以节点的方式体现出来。

Sax方式是流式解析，一点一点读文档。

XML和JSON对比

XML易读，但是数据量大

JSON数据量小。目前移动端应用普遍采用。

29、类别用的多不多？类别你都是怎么用？类别都有什么作用？（类别、类目）

参考回答：用过。类别主要就是在原有类的基础上，不通过继承的方式，添加新的方法，分散类的实现，方法的私有化。

1. 给系统原有的类添加方法，不能直接扩展属性，系统强烈不建议，覆盖原有类的方法，如果想覆盖原有类的方法，通过继承实现。
2. 分散类的实现：如＋（NSIndexPath ＊）indexPathForRow：（NSInteger）row inSection：（NSInteger）section；原本是属于NSIndexPath的方法，但是因为这个方法经常在使用表的时候来调用，跟表的关系特别密切。所以把这个方法以类别的形式，声明在了UITableView.h中。
3. 声明私有方法。不在.h中声明方法，只在.m中声明一个类别方法。

我们可以通过“关联” Associated的技术，变相的达到给类增加属性的目的。实际上是利用了oc的运行时编程，把一个变量绑定到一个对象上。

30、类别和扩展有什么区别？

扩展写法上跟类别一致，只是括号中没有类别描述。

“扩展”可以添加属性、变量，类别不能。

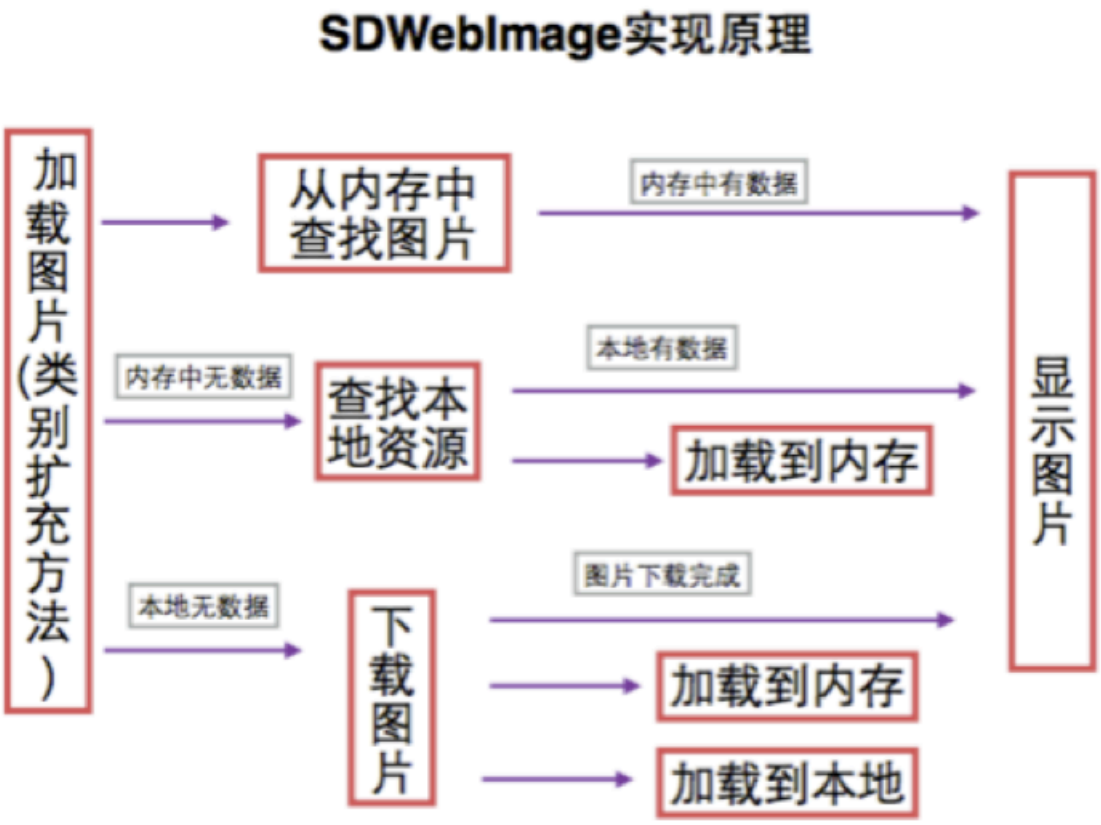
31、如何清理图片缓存?

因为随着应用程序不断缓存图片，为造成硬盘压力变大，比如网易新闻中，每天看新闻，每天都缓存图片，日积月累，占用空间越来越大，所以一般会在软件设置界面中，提供一个清除缓存的功能。实际面试过程中，可以把网易新闻替换为一个自己做的项目。

调用SDWebImageCache的 clear（清除所有缓存），clean（清除过期缓存）方法。

32、SDWebImage的实现原理是什么?

调用类别的方法：1.从内存（字典）中找图片（当这个图片在本次使用程序的过程中已经被加载过），找到直接使用。2.从沙盒中找（当这个图片在之前使用程序的过程中被加载过），找到使用，缓存到内存中。3.从网络上获取，使用，缓存到内存，缓存到沙盒。



33、iOS中你使用过哪些设计模式，你是怎么理解的?

单例模式：（需要能完整的写出单例的实现，理解为什么要使用线程同步，要覆盖系统的哪些方法，方法内的具体实现，为什么要使用单例模式）
使用单例模式的场景：全局公共的数据，每个类中都有可能要获取同一份数据，可以存储到单例类的属性中，这样每个类都可以方便的访问同一份数据。（比如全局的设置类，用户的登陆信息，想一想系统都有哪些单例类，为什么他们是单例类）

代理模式：解决类和类之间的事件、值传递。把一个类中发生的事件通知到另一个类中，使用代理模式可以降低类和类之间的耦合度。

观察者（通知、KVO）：通知和代理解决的问题类似，但是相比代理，有以下几处不同

1.代理一对一，通知一对多

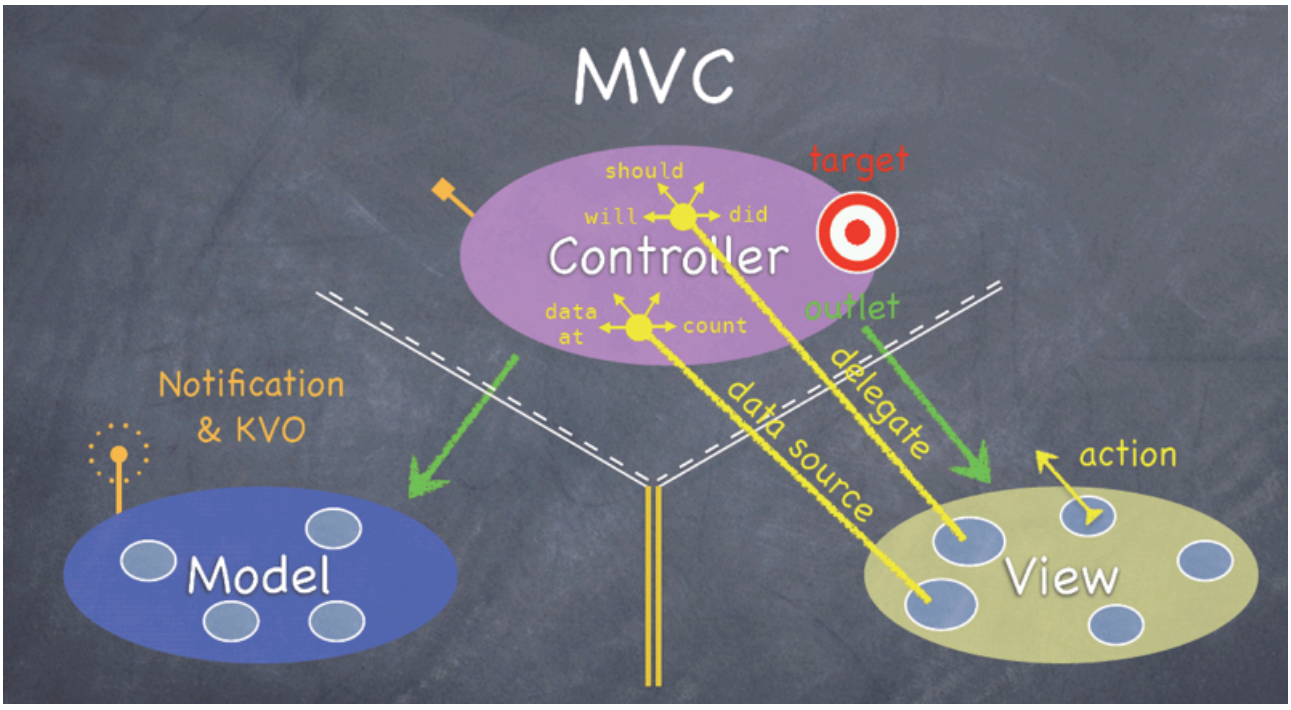
2.代理可以相互传值，通知只能是单向传值，发通知的对象给接收通知的对象传值

3.通知的效率低于代理。（想一想为什么，通知是需要查询所有注册者的信息的，符合接收条件，才调用对象的方法，代理是直接对象调用方法。回顾通知和代理的原理，你就懂了）。

4.代理是有线（有线网络）的，通知是无线（wifi）的。

工厂模式：工厂模式解决的问题是多态，一个类可能有多个子类，具体需要哪个子类，工厂根据不同的条件返回不同的子类对象。在iOS中类簇就是工厂模式。（使用类簇的类，NSString，NSNumber，NSArray等）。

34、MVC是什么，你对MVC的理解?



MVC总体来说解决的问题就是类和类之间的耦合度降低问题，类和类的耦合度降低有利于后期的代码修改，代码扩展，代码维护，代码排错。（就跟衣柜里面的衣服分门别类的放置一样的道理，袜子不会和内裤放一块，分开放比较好）

MVC是一种架构模式，M表示数据模型Model V表示视图View C表示控制器Controller。

1.Model负责存储，定义，操作数据。

2.View用来展示数据给用户，和用户进行操作交互。

3.Controller是Model与View的协调者，Controller把Model中的数据拿过来给View用。

Controller可以直接与Model和View进行通信，而View不能和Controller直接通信。View与Controller通信需要利用代理协议的方式，当有数据的更新时，Model也要与Controller进行通信，这个时候就用notification和KVO，这个方式就像一个广播一样，Model发信号，Controller设置监听器接受信号，当有数据要更新时，就发信号给Controller。Model和View不能直接进行通信，因为这样就违背了MVC的设计思想。

35、你对KVC/KVO的理解?

KVC

Key value Coding是cocoa的一个标准组成部分，它能让我们可以通过name（key）的方法访问property，不必调用明确的property accesser（set/get方法）。

KVC(键－值编码)是一个用于间接访问对象属性的机制（一种使用字符串而不是访问器方法去访问一个对象实例变量的机制），使用该机制不需要调用set或者get方法以及－>来访问成员变量，它通过setValue：forkey：和valueForkey：方法。

KVC的机制是啥样的呢? 它是以字符串的形式向对象发送消息字符串是要关注属性的关键。
是否存在setter，getter方法，如果不存在，它将在内部查找名为_key或key的实例变量，如果没有会调用setValueForUndefindedKey：，如果也没有，则会运行时报错;
注意：如果是基本数据类型，则需要封装一下（NSNumber）。

KVC的使用环境：无论是property还是普通的全局属性变量，都可以用KVC。

KVC的优缺点:
优点：1.主要的好处就是来减少代码量 2.没有property的变量（private）也能通过KVC来设置;
缺点：如果key写错时，编写时不会报错，运行时会报错;

KVO

KVO是一个对象能够观察另外一个对象的属性值，并且能够发现值的变化。KVO更加适合任何类型的对象侦听另外一个任意对象的改变，或者是一个对象与另外一个对象保持同步的一种方法，即当另外一种对象的状态发生改变时，观察对象马上作出反应。它只能用来对属性作出反应，而不会用来对方法或者动作作出反应。

KVO的优点：

1.能够提供一种简单的方法实现两个对象间的同步；

2.能够对非我们创建的对象，即内部对象的状态改变作出响应，而且不需要改变内部对象（SDK对象）的实现；

3.能够获得观察的属性的最新值以及先前值；

4.用key path来观察属性，因此也可以观察嵌套对象（也就是可以观察一个对象内部对象的属性的变化，可以无限嵌套观察,前提是对象的属性支持KVO）；

5.完成了对观察对象的抽象，因为不需要额外的代码来允许观察值能够被观察（不需要像通知一样还需要发送通知，KVO属性的改变，外部可以直接观察）。

KVO的注意事项：

我们注册KVO的时候，要观察哪个属性，在调用注册方法的时候，addObserver：forKey：options：context： forKey处填写的属性是以字符串形式，万一属性名字写错，因为是字符串，编译器也不会出现警告以及检查；

KVO的使用：被观察者发出addObserver：forKey：options：context：方法来添加观察者。然后只要被观察者的keyPath的值变化（注意：单纯改变其值不会调用此方法，只有通过getters和setters来改变值才会触发KVO），就会在观察者里调用方法observeValueForKeyPath：ofObject：change：context：因此观察者需要实现方法observeValueForKeyPath：ofObject：change：context；来对KVO发出的通知做出响应。

这些代码只需要在观察者里进行实现，被观察者不用添加任何代码，所以谁要监听谁注册，然后对响应进行处理即可，使得观察者与被观察者完全解藕，运用很灵活很简便；但是KVO只能检测类中的属性，并且属性名都是通过NSSTring来查找，编译器不会帮你检错和补全，纯手敲所以比较容易出错。

36、多线程的实现方式有哪些？并做对比

NSThread

相当于自己创建一个线程，创建线程的时候，可以把一个方法放到创建的线程中。

创建方式主要有两种：

1. [NSThread detachNewThreadSelector:@selector(myThreadMainMethod:) toTarget:self withObject:nil];

2. NSThread* myThread = [[NSThread alloc] initWithTarget:self
selector:@selector(myThreadMainMethod:) object:nil];
[myThread start]; //启动线程

优点：NSThread 比其他两个轻量级。
缺点：需要自己管理线程的生命周期，线程同步，线程同步时对数据的加锁会有一定的系统开销。

NSOperation （使用NSOperation和NSOperationQueue）
NSOperation 不需要自己创建线程，只关注需要在分线程中完成的代码，然后把NSOperation 放到NSOperationQueue中即可，NSOperationQueue会把代码放到分线程中执行。

NSOperation的作用：配合使用NSOperation和NSOperationQueue也能实现多线程编程。
NSOperation和NSOperationQueue实现多线程的具体步骤：
（1）先将需要执行的操作封装到一个NSOperation对象中的main方法
（2）然后将NSOperation对象添加到NSOperationQueue中
（3）系统会自动将NSOperationQueue中的NSOperation取出来
（4）将取出的NSOperation封装的操作放到一条新线程中执行

NSOperation的子类
NSOperation是个抽象类,并不具备封装操作的能力,必须使用它的子类。
使用NSOperation子类的方式有3种：
（1）NSInvocationOperation
（2）NSBlockOperation
（3）自定义子类继承NSOperation,实现内部相应的方法

优点：不需要关心线程管理， 数据同步的事情，可以把精力放在自己需要执行的操作上。

GCD（Grand Central Dispatch）
Grand Central Dispatch 简称（GCD）是苹果公司开发的技术，以优化的应用程序支持多核心处理器和其他的对称多处理系统的系统。这建立在任务并行执行的线程池模式的基础上的。

GCD的工作原理：
-> 让程序平行排队的特定任务，根据可用的处理资源，安排他们在任何可用的处理器核心上执行任务。
-> 一个任务可以是一个函数(function)或者是一个block。GCD的底层依然是用线程实现，不过这样可以让程序员不用关注实现的细节。

-> GCD 中必须要使用的是各种队列，我们通过block，把具体的代码放到队列中，队列中的任务排队执行，系统会自动的把队列中的各个任务分配到具体的线程中和cpu中（如果是多核处理器），具体创建多少个线程，分配到哪个cpu上，都是由系统管理。

GCD中有三种队列类型:

• The main queue: 系统自带的一个队列，放到这个队列中的代码会被系统分配到主线程中执行 。main queue可以调用dispatch_get_main_queue()来获得。因为main queue是与主线程相关的，所以这是一个串行队列，提交至其中的任务顺序执行 （一个任务执行完毕后，再执行下一个任务）。

• Global queues: 整个应用程序存在三个全局队列（系统已经创建好，只需要获得即可）：高、中（默认）、低三个优先级队列。可以调用dispatch_get_global_queue函数传入优先级来访问队列。全局队列是并行队列，可以让多个任务并发（同时）执行（自动开启多个线程同时执行任务）并发功能只有在异步（dispatch_async）函数下才有效。

• 用户自己创建队列: dispatch_queue_create 创建的队列. 可以是串行的，也可以是并行的，因为系统已经给我们提供了并行，串行队列，所以一般情况下我们不再需要再创建自己的队列。用户创建的队列可以有任意多个。

注意：分线程中不能刷新UI,刷新UI只能在主线程。如果每个线程都可以刷新UI，将会很容易造成UI冲突，会出现不同步的情况，所以只有主线程中能刷新UI系统是为了降低编程的复杂度，最大程度的避免冲突。

分线程中回到主线程主要有两种方式

1.performSeletorOnMainThread

2.使用main queue

分线程在使用的时候，有以下几个需要说明的地方

1. 之前的版本中分线程不会自动创建autorelease pool，所以需要在分线程创建autorelease pool。目前的sdk版本已经不需要。

2. timer不能在分线程中直接使用，需要手动开启run loop。

3. 如果多个线程修改（只是读取变量，不会有问题）同一个资源，需要注意线程同步的问题。

37、timer的间隔周期准吗？为什么？怎么样实现一个精准的timer?

定时器timer一般都是准确的，但是当主线程有些时候难免会出现堵塞情况，这样就可能导致定时器timer会延迟从而不准确，

我们可以开一个多线程，在多线程上运行定时器，这样多线程只运行定时器，不会堵塞而导致误差。

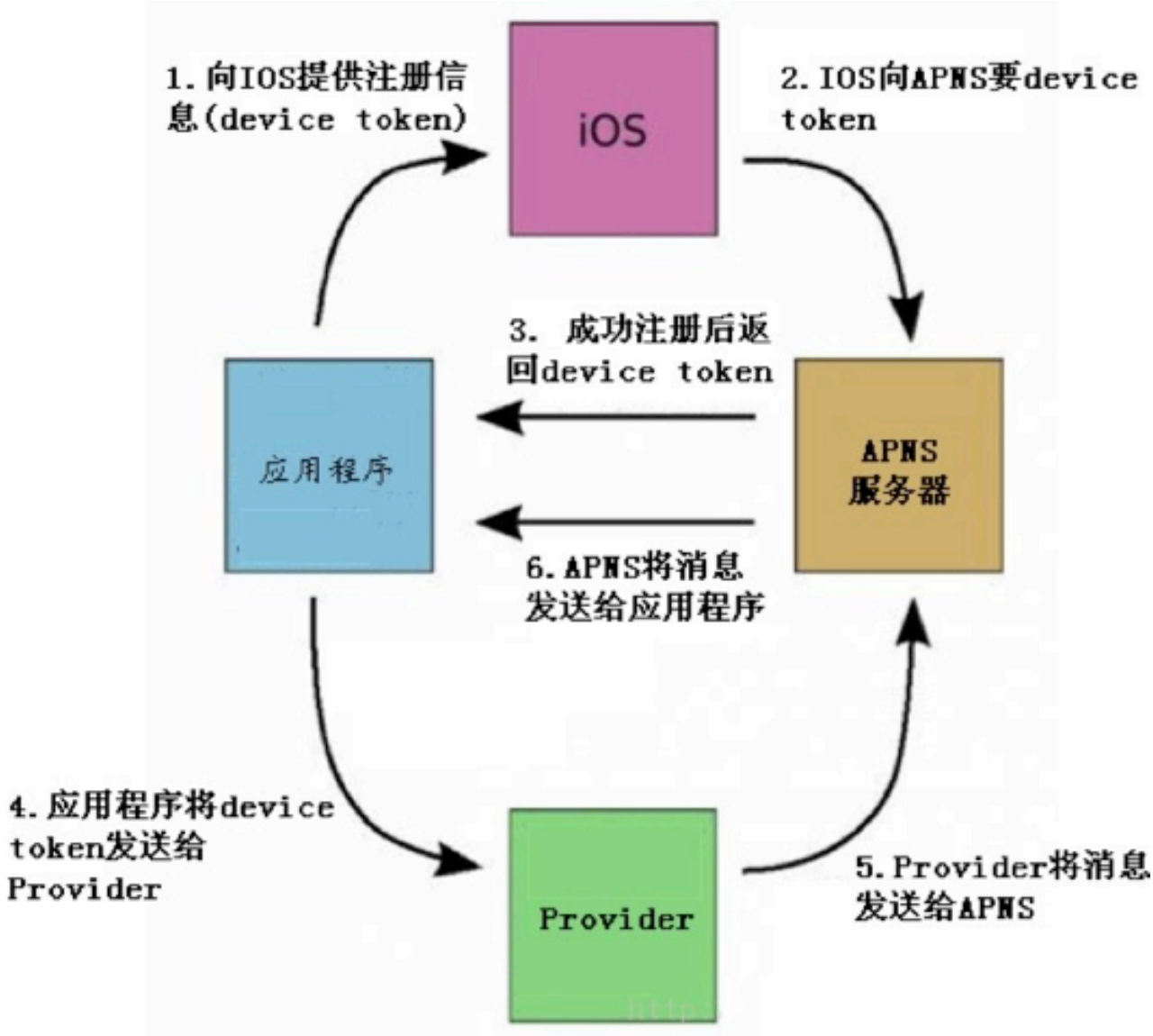
38、你经常使用哪些第三方库？常用的第三方库有哪些？

1.ASIHTTPRequest，AFNetworking　　2.FMDB　　3.SDWebimage

4.SVPullToReresh　　5.DDMenu　　6.ZBar

7.MBProgressHUD　　8.Masonry　　等

39、如何进行网络推送？推送的流程是什么？你的推送是怎么做的？（网络推送、本地推送）



推送分为本地推送以及网络推送

网络推送APNS分为注册部分以及推送部分

注册部分

1.我们的应用向我们的系统注册推送

2.我们的系统会向用户询问是否允许推送

3.用户允许后，我们系统会向苹果推送服务器索要device token。

4.苹果推送服务器会将device token返回给我们客户端。

5.我们应用将device token发送给我们服务器。

推送部分

1.我们服务器将推送的消息，以及device token发送给苹果推送服务器

2.苹果推送服务器会讲推送的消息内容发送给我们客户端。

推送注意事项：

1.ios7和ios8推送不同。

2.如果指定推送声音本地必须有这个文件，如果声音为系统默认声音。

3.服务器需要配置cer证书，证书生成时，需要CSR证书签名请求。

4.如果用户选择不允许推送，那么无法进行推送。

5.推送是免费的。

6.推送是不可靠的，用户不一定能立即收到该消息。

7.推送消息大小有限制，长度256字节

注意：也有一些平台提供第三方的推送服务，但实际上都是基于苹果推送本身的，只是对苹果本身的推送封装了一下。

40、CALayer和UIView之间的区别与联系?

每一个UIView都包含一个CALayer对象，CALayer中存储的是UIView的展示内容数据，负责绘制。UIView管理CALayer，相当于一个管理者，并具备处理触摸事件的能力（因为uiview继承自uiresponder）。

41、核心动画里面有哪些常用的动画类?

常用的动画类：CABasicAnimation基础动画；CAKeyframeAnimation关键帧动画；CAAnimationGroup动画组；CATransition转场动画。

42、地图中iOS8定位失败你是如何解决的?

需要在info.plist文件里面加NSLocationWhenInUseDescription或NSLocationAlwaysUsageDescription

43、常用的手势有哪些？如何自定义手势?

系统提供了七种手势帮我们进行手势的识别开发：点击UITapGesture，滑动UIPanGesture，轻扫UISwipeGesture，旋转UIRotationGesture，缩放UIPinchGesture，长按UILongPressGesture，屏幕边缘滑动UIScreenEdgePanGestureRecognizer，

自定义手势：可以在touchesBegin ，touchesMove，touchesCanceled，touchesEnd中写自己的手势逻辑.如果再细问，就说一般系统的手势已经够用，很少自定义。

44、推送失败你是如何解决的?

iOS8之后的代码逻辑稍有改变。

```
#ifdef __IPHONE_8_0
   //IOS8执行以下代码
   [[UIApplication sharedApplication] registerForRemoteNotifications];

   [[UIApplication sharedApplication] registerUserNotificationSettings:[UIUserNotificationSettings
settingsForTypes:UIUserNotificationTypeBadgeIUIUserNotificationTypeSoundI
UIUserNotificationTypeAlert categories:nil]];
#else
   //IOS7执行以下代码
   [[UIApplication sharedApplication] registerForRemoteNotificationTypes:
(UIRemoteNotificationTypeAlert I UIRemoteNotificationTypeSound I
UIRemoteNotificationTypeBadge)];
#endif
```

45、iOS9 http失败，你是如何解决的?

1. 在Info.plist中添加NSAppTransportSecurity类型Dictionary。
2. 在NSAppTransportSecurity下添加NSAllowsArbitraryLoads类型Boolean,值设为YES。

46、地图定位偏移你该怎么办？（火星坐标）

火星坐标系统

火星坐标系统是一种国家保密处理，其实就是对真实坐标系统进行人为的加偏处理，按照特殊的算法，将真实的坐标加密成虚假的坐标，而这个加偏并不是线性的加偏，所以各地的偏移情况都会有所不同。而加密后的坐标也常被人称为火星坐标系统。

解决方式一：在GPS软件中设置一个使用同样算法的加偏移功能，即：GPS先从卫星上得到真实坐标，然后经过加偏移程序转换成火星坐标，由于是同一个算法，所以经过软件加偏移的坐标能跟同

样加了偏移的地图吻合，一般不使用，太麻烦，也不知道转换算法，这种方式是专业制作地图的公司可能会使用。

解决方式二：我们使用第三方地图的时候，第三方地图肯定已经处理过，直接用系统的CLLocationManager获得坐标是真实没有处理的，但是第三方地图中，如百度地图，传入的坐标要经过处理，所以最好的方式就是 定位坐标的代码，和显示地图的代码使用同一个平台的代码。例如如果你使用百度地图，就不要使用CLLocationManager来获得坐标，再传入百度地图，这样地图就会出现偏移。应该使用BMKLocationService来获得坐标，再传入百度地图，这样才能显示正确（不能一国两制）。

47、OC的消息机制你知道不知道?

如果记不住下面的理解：只需要回答，系统内部是靠objc_msgSend来实现方法调用的。

在Objective-C中，message与方法的真正实现是在执行阶段绑定的，而非编译阶段。编译器会将消息发送转换成objc_msgSend方法的调用。objc_msgSend方法含两个必要参数：receiver、方法名（即：selector）。如：[receiver message]；将被转换为：objc_msgSend(receiver selector);
objc_msgSend方法也能hold住message的参数，如: objc_msgSend(receiver,selector,arg1,arg2,…);
objc_msgSend方法会按照顺序进行以下操作，以完成动态绑定：查找selector所指代的程序（方法的真正实现）。因为不同类对同一方法有不同的实现，所以对方法的真正实现的查找依赖于receiver的类调用该实现，并将一系列参数传递过去将该实现的返回值作为自己的返回值，返回之消息传递的关键是，编译器构建每个类和对象所采用的数据结构、每个类都包含以下两个必要元素：一个指向父类的指针一个调度表（dispatch table）。该调度表将类的selector与方法的实际内存地址关联起来。每个对象都有一个指向所属类的指针isa。通过该指针，对象可以找到他所属的类，也就找到了其全部父类。

48、OC的事件响应者链

响应者链表示一系列的响应者对象。事件被交由第一响应者对象处理，如果第一响应者不处理，事件被沿着响应者链向上传递，交给下一个响应者（next responder）。一般来说，第一响应者是个视图对象或者其子类对象，当其被触摸后事件被交由其他处理，如果他不处理，事件就会被传递给它的试图控制器对象（如果存在）。然后是他的父视图（superview）对象（如果存在），以此类推，直到顶层视图。接下来会沿着顶层视图（top View）到窗口（UIWindow对象）再到程序（UIApplication对象）。如果整个过程都没有响应这个事件，该事件就被丢弃。一般情况下，在响应者链中只要由对象处理事件，事件就停止传递。但有时候可以在试图的响应方法中根据一些判断条件来决定是否需要继续传递事件。

49、常用的数据结构都是有哪些?

数据结构是计算机存储、组织数据的方式。如数组，字典都是一种数据结构，他们的组织方式和原理就不一样。下面说的这些都是比较理论的（知道名字就ok），在不同的语言中，可能有不同的实现方式，例如在iOS中NSDictionary就是类似hash表的一种结构。

1.栈：　　先进后出，相当于一个没有盖子的杯子。

2.队列:　　先进先出，相当于一个水管。Fifo（first in first out）

3.二叉搜索树

4.散列表（hash表）

5.检索树（Trie）

6.优先队列

7.线段树和树状数组

8.后缀树与后缀数组

9.并查集

10.邻接表和边表

数据元素相互之间的关系称为结构。有四类基本结构：**集合、线性结构、树形结构、图状结构;**

集合结构：除了同属于一种类型外，别无其它关系

线性结构：元素之间存在一对一关系常见类型有: 数组,链表,队列,栈,它们之间在操作上有所区别.例如:链表可在任意位置插入或删除元素,而队列在队尾插入元素,队头删除元素,栈只能在栈顶进行插入,删除操作.

树形结构：元素之间存在一对多关系,常见类型有:树(有许多特例:二叉树、平衡二叉树、查找树等)

图形结构：元素之间存在多对多关系,图形结构中每个结点的前驱结点数和后续结点数可以任意多个。

50、如何将产品进行多语言发布，做国际化开发?

下面是步骤，可以简洁的回答为，把程序内需要用的字符串，写为多个语言版本，程序内通过NSLocalizedString宏定义来读取，系统会根据本机语言，来读取对应的语言文本。

1、新建String File文件，命名为Localizable.strings,往里面添加你想要的语言支持。

2、在不同的语言的Localizable.strings文件中添加对应的文本。

3、XIB国际化。在需要国际化的XIB文件上添加get Info添加多语言版本，修改个语言版本相对应的界面文字及图片。

4、程序名称国际化。新建一个syrings文件，然后国际化他，get Info…。

51、OC的RunLoop

RunLoop是线程相关的基础框架的一部分。一个run loop就是一个事件处理的循环，用来不停的调度工作以及处理输入事件。使用run loop的目的是让你的线程在有工作的时候忙于工作，而没工作的时候处于休眠状态。

52、平时做项目中屏幕适配你是怎么做的?

一般不直接使用autolayout，系统api比较麻烦，一般使用**Masonry**。

Masonry本质上是对autolayout的封装。

用法：大的页面关系，用计算完成；每个小块里面的相对位置关系，用Masonry来做。在有些场景下，Masonry有非常大的优势。

比如说：1、设置某个View的宽高比　　2、设置居中，设置相对边距

53、iOS系统的版本比较多，你是如何适配的?

一般可以通过系统的宏定义或者UIDevice中 的systemVersion来判断系统的版本。

如果低版本和高版本中要实现同一个功能，api或者处理流程不一样，会使用条件编译来处理，把低版本，高版本的代码都写出来，根据不同的系统版本，选择性的来编译不同的代码。

54、iOS上应用如何兼容32位系统和64位系统?

在苹果推出iPhone5S时，cpu采用64位，之前的手机采用32位，所以就有了兼容的问题，而且苹果已经规定，以后上传的应用必须支持64位。

让应用兼容64位的基本步骤

在Xcode中打开工程 ， 在Project Setting里面，把最小应用使用的SDK改到5.1.1或者更高的版本，把build setting中的Architectures参数设置成“Standard Architectures (including 64-bit).”

这样你的应用就支持了64位的CPU，再次修复编译器的错误和警告。

注意：改完之后，如果你的程序内使用是int，float，那么可能会有隐患，因为在64位的系统上，所占用的字节数可能跟32位系统上的不一致，最好使用NSInteger来代替int，CGFloat代替float，这样系统可以自动替换合适的类型。

应用在兼容64位系统后，内存的占用肯定会变多一点（因为安装包中包含了32位和64位的两套指令），不过性能也有相应的提升。

55、storyboard用过吗？有什么特点？你写项目的过程中使用的是纯代码还是storyboard，有什么优缺点？

Storyboard是iOS5新增的特性，是对XIB的升级版本，引入了一个容器用于管理多个视图控制器的XIB文件，和视图控制器之间的跳转交互。

优点：不用再为每个控制器创建XIB文件了，可以使用静态cell，当cell固定且不多时，使用起来比较方便。

缺点：Storyboard单个文件，不利于团队协作开发。

56、你发布的应用程序你是如何捕获异常的?

目的是：当你的应用程序被用户安装后，用户在使用的过程中，把程序的崩溃日志存储到本地，并且在合适的时候发送给开发者，达到收集崩溃日志的目的，这不是必须的，只是一般会这样做，这些数据可以用于我们分析崩溃日志，修改bug。

有一些第三方库可以实现，如友盟等，但是我们公司开发的项目中一般都没有用。

57、你做过二维码没？做二维码的时候你是如何用的?

扫描二维码的开源库有很多如 ZBar、ZXing等。iOS7之后，系统提供的也有，但是不好用。

58、有没有做过支付？你是怎么做的?

参考回答：以前的项目中还没有接触过，但是自己了解一点，需要到支付宝或者微信的开放平台申请，开放平台内有详细的对接文档。

支付宝支付大概步骤：

开放平台 (找不到iOS sdk)

http://open.alipay.com/index.htm(这个里面找不到sdk) 需要进入下面的链接

现在不少app内都集成了支付宝功能

使用支付宝进行一个完整的支付功能，大致有以下步骤：

1>先与支付宝签约，获得商户ID（partner）和账号ID（seller）

(这个主要是公司的负责)

2>下载相应的公钥私钥文件（加密签名用）

3>下载支付宝SDK

官方sdk页面地址：(藏得很深)

https://b.alipay.com/order/productDetail.htm?productId=2014110308141993&tabId=4#ps-tabinfo-hash

里面提供了非常详细的文档、如何签约、如何获得公钥私钥、如何调用支付接口。

4>生成订单信息

5>调用支付宝客户端，由支付宝客户端跟支付宝安全服务器打交道

6>支付完毕后返回支付结果给商户客户端和服务器

SDK里有集成支付宝功能的一个Demo> 集成支付功能的具体操作方式，可以参考Demo

微信支付：微信开放平台https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=1417694084&token=&lang=zh_CN

59、有没有做过即时聊天，你是怎么实现的?

参考回答：没有做过，但是知道其中的原理。实现即时聊天要使用socket来实现，但是鉴于自己实现的复杂度，一般情况下都是使用第三方平台的服务，例如环信，爱萌等，但是坏处是数据在他们的平台存放。我们也可以使用XMPP协议，自己写，也有现成的一些框架，这样相对简单，数据又可以在自己的服务器上存放，只是相比使用环信等，开发周期和难度上会适当加大。

注意：XMPP中数据交换使用的是XML，不是JSON。

60、你有没有使用过swift?

了解过一点，但是我们公司开发的时候还是使用oc，平常业余时间自己学习过一点。

Swift的代码更简洁，官方说运行效率会更高。但是一般公司不用的原因有以下几点：

1.转换成本。公司内的软件工程师都已经习惯oc，都换做swift来开发，那么势必降低开发效率，增加工程师的学习压力，而且还有可能因为大家都刚开始用swift，造成写出来的程序不稳定，作为公司来说，要的是出产品的速度和产品的稳定，而不是纠结采用哪门语言开发。

2.第三方库：大部分第三方库还是使用oc来开发的。

3.iOS7不支持swift，那样我们如果使用swift，开发的程序将无法运行在iOS7。

除非苹果出大招，不使用swift编写的程序无法提交app store，这样才会逼迫大家学习并且使用swift，不然短时间内不大可能普及。

61、你使用过的版本控制工具（团队协作工具）有哪些?

我们公司用的svn。

如何防止多人修改一个文件，发生冲突：

1.一般责任划分的时候，尽量避免多人修改同一个文件的情况

2.如果要修改，先给其他同事说一声，等你提交了，其他同事再修改。

2.如果真冲突了，应该先更新，再把自己的内容修改过后，重新提交。

如果问是否使用过git，回答没有，了解过一点。

62、如何打包静态库?

新建一个Framework&Library的项目，编译的时候，会将项目中的代码文件打包成一个.a的静态库文件。

编译的时候分为device（arm版本）版本和模拟器版本（i386，x86_64）。

63、如果使用svn静态库如何提交?

直接提交不上去，一般通过命令行手动提交上去（如：svn add libzbar.a），一般svn工具是默认无法提交.a 文件的。

64、如何进行真机调试（经验性的题）

真机调试步骤:

1.登录http://developer.apple.com/。

2.使用钥匙串访问生成CSR文件。

3.上传CSR文件生成ios_development.cer证书.

4.添加appId，也就是Bundle ID

5.添加设备ID，也就是UDID。

6.选择appID、cer证书、设备,生成mobileprovision也就是齿轮文件

7.打开cer证书以及mobileprovisin齿轮文件就可以真机调试了。

真机调试的注意事项:

1.真机调试的设备必须在勾选的设备id里面。

2.当前Xocde最高支持ios系统>=手机版本系统>=应用程序最低支持IOS系统

3.appID与应用Bundle ID必须一致，除非添加appid时，最后以 * 结尾，比如com.zhiuyou.*。

65、APP发布上架流程，你如何打包app（生成ipa）?

上架流程

1.通过https://ituesconnect.apple.com添加应用信息。

2.下载安装发布证书。

3.选择发布证书，使用Archive编译发布包。用Xcode将代码（发布包）上传到服务器。

4.等待审核通过。

生成ipa：通过菜单栏Product 下的 Archive生成ipa文件。