

iOS 面试题

1.OC语言的基本特点

OC语言是C语言的一个超集，只是在C的基础之上加上了面向对象（oo）的特性；
OC与Java语言相同都是单继承，这一点与C++语言不同（多重继承）；
OC不支持命名空间机制，取而代之的是在类名之前添加前缀，以此来区分。

2.#include与#import的区别、#import与@class的区别

(1) #include与#import的区别：#include和#import其效果相同，只是后者不会引起交叉编译，确保头文件只会被导入一次。
(2) #import与@class的区别：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，至于这些类是如何定义的，暂时不用考虑，后面会再告诉你。使用#import编译效率高，防止相互包含的编译错误！

3.简述类目优点和缺点，如果覆盖本类或者父类的方法，会出现什么问题？

(1)优点：不需要通过增加子类而增加现有类的行为（方法），且类目中的方法与原始类方法基本没有区别
(2)缺点：无法向类目添加实例变量。覆盖原始类方法后，原始类的方法没办法调用

4.简述内存管理基本原则

如果使用alloc、copy（mutableCopy）或者retain一个对象时，你就有义务，向它发送一条release或者autorelease消息。其他方法创建的对象，不需要由你来管理内存。

5. 什么是ARC技术？与GC是否相同？

ARC是Automatic Reference Counting的简称，我们称之为自动引用计数，是iOS5.0之后推出的内存管理的新特性。本质上还是使用引用计数来管理对象，只是我们在编写代码时，不需要向对象发送release或者autorelease方法，也不可以调用dealloc方法，编译器会在合适的位置自动给用户生成release消息（autorelease）。GC的全称是garbage collection，内存垃圾回收机制，ARC比GC性能好！

6.iOS中是否支持垃圾回收机制

iOS开发只支持手动内存管理与ARC，Mac开发支持GC垃圾回收机制，10.8之后弃用了GC，推荐使用ARC。

7.深、浅复制的基本概念以及他们的区别

浅复制：只复制对象本身，不对里面的属性复制

深复制：不仅复制对象本身，对象持有的属性对象也做复制

8.内存中的堆区和栈区的差别

栈区（stack）由编译器自动分配释放，存放方法（函数）的参数值，局部变量的值等，堆区（heap）一般由程序员分配与释放，若程序员不释放，则内存溢出

9.用户自定义了一个对象，如何实现拷贝（可变和不可变拷贝）

必须实现copying和mutableCopying协议，表示返回一个不可变和可变的对象。否则，程序将会出现异常。

```
- (id)copyWithZone:(NSZone *)zone
{
    Person *person = [[self Class] allocWithZone:zone];
    person ->age = self.age;
    person ->name = self.name;
    return person;
}
- (id)mutableCopyWithZone(NSZone *)zone;
```

10.定义属性时，什么时候用assign、retain、copy、nonatomic

(1)assign：普通赋值，一般常用于基本数据类型，常见委托设计模式，以此来防止循环引用。

(2)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。

(3)用来复制对象，一般字符串使用copy，Foundation中的不可变对象使用copy效果相当于retain，只是引用计数+1

(4) nonatomic，非原子性访问，不加同步，多线程并发访问会提高性能

11. strong与weak，_unsafe_unretained与weak的区别

(1)strong叫强引用，weak叫弱引用，在ARC中，使用strong告诉编译器帮助我们自动插入retain，weak是普通赋值相当于手动管理内存的assign，

(2)_unsafe_unretained与weak功能一致，区别在于当指向的对象销毁后，weak会将变量置为nil,防止调用野指针。

12.ARC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露

13.当我们释放对象时，为什么需要调用[super dealloc]方法？

因为，子类是继承自父类，那么子类中有一些实例变量（对象），是继承自父类的，因此，我们需要调用父类方法，将父类所拥有的实例进行释放。

14.自动释放池是什么，如何工作

自动释放池是NSAutorelease类的一个实例，当向一个对象发送autorelease消息时，该对象会自动入池，待池销毁时，将会向池中所有对象发送一条release消息，释放对象

15.为什么delegate（代理）属性都是assign而不是retain的？

防止循环引用，以至对象无法得到正确的释放

16.iOS开发中数据持久性，有哪几种

plist文件写入、对象归档、sqlite3数据库、coredata

17.什么是KVC和KVO？以及它们之间的关系是什么

(1)KVC（键值编码）是一种间接访问对象实例变量的机制，该机制可以不通过存取方法就可以访问对象的实例变量

(2)KVO（键值观察）是一种能使得对象获取到其他对象属性变化的通知机制。

(3)实现KVO键值观察模式，被观察的对象必须使用KVC键值编码来修改它的实例变量，这样才能被观察者观察到。因此，KVC是KVO的基础或者说KVO的实现是建立在KVC的基础之上的。

18. 简述常见的设计模式

单例设计、代理设计、观察者(通知)、工厂方法、模版方法、

19. 内存管理 在dealloc放发中 用release方法号还是用self.xxx = nil 好；

使用self.xxx=nil更好，因为先调用了release方法，而且还将变量设置为nil，这样就更安全的释放对象，防止野指针调用

20.线程与进程的区别和联系？

一般的应用程序是单个进程，也有多进程(谷歌浏览器)，进程是个静态的容器，里面容纳了很多个线程，线程是一系列方法的线性执行路径。

21.OC语言的优缺点

(1)优点：类目、动态识别、支持C语言、OC与C++可以混编

(2)缺点：不支持命名空间、不支持运算符重载、不支持多重继承

22.代理delegate与通知Notification、block的使用区别

delegate与block一般是用于两个对象1对1之间的通信交互，delegate需要定义协议方法，代理对象实现协议方法，并且需要建立代理关系才可以实现通信。block更加简洁，不需要定义繁琐的协议方法，但是如果通信事件比较多的话，建议使用delegate。Notification主要用于1对多情况下通信，而且，通信对象之家不需要建立关系！但是使用通知，代码可读性差

23. 控制器 **ViewController** 的 **loadView**, **viewDidLoad**, **viewDidUnload** 分别是在什么时候调用的?

- (1) **loadView**调用: 当控制器的根视图**view**为空, 且此**view**被访问时则调用
- (2) **loadView**调用之后调用**viewDidLoad**
- (3) **iOS6.0**之前, 当内存警告时, 先卸载视图, 再调用**viewDidUnload**来释放内存

24. **@synthesize**与**@dynamic**的区别

- (1) **@synthesize** 是系统自动生成**getter**和**setter**属性声明
- (2) **@dynamic**的意思是告诉编译器, 属性的获取与赋值方法由用户自己实现, 不自动生成。

25. 事件响应者链的概念

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它处理, 如果它不处理, 事件就会被传递给它的视图控制器对象 (如果存在), 然后是它的父视图 (**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。但有时候可以在视图的响应方法中根据一些条件判断来决定是否需要继续传递事件。

26. 线程与进程的区别

27. C语言中**static**变量和**static**函数有什么作用

- (1) 表示变量是静态存储变量, 表示变量存放在静态存储区
- (2) 加在函数前面的时候表示该函数是内部连接, 只在本文件中有效, 别的文件中不能应用该函数。

28. 指针题

- a) `int a;` // 一个数值
- b) `int *a;` // 一个指向数值的指针
- c) `int **a;` // 一个指向指针的指针
- d) `int a[10];` // 一个有10个指针的数组, 该指针是指向一个整型数的。
- e) `int *a[10];` // 一个有10个指针的数组, 该指针是指向一个整型数的。
- f) `int (*a)[10];` // 一个指向有10个整型数数组的指针
- g) `int (*a)(int);` // 一个指向函数的指针, 该函数有一个整型参数并返回一个整型数。
- h) `int (*a[10])(int);` // 一个有10个指针的数组, 该指针指向一个函数, 该函数有一个整型参数并返回一个整型数。

29. 写一个**NSString**类的实现

```
+ (id)initWithCString:(const char *)nullTerminatedCString encoding:
(NSStringEncoding)encoding{
    NSString *obj;
    obj = [self allocWithZone: NSDefaultMallocZone()];
    obj = [obj initWithCString: nullTerminatedCString encoding: encoding];
    return AUTORELEASE(obj);
}
```

30.c和obj-c如何混用，C++和obj-c呢

实现文件的扩展名.m改成.mm即可！但cpp文件必须只能用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

31.obj-c中有私有方法、私有变量吗？

(1)使用private修饰的全局变量是私有的变量

(2)obj-c在语法上没有设计私有的方法，不过可以通过延展在实现文件里定义方法作为私有的方法。但不是真正意义上的私有方法，也是可以调用的，只是外部不可见而已。

32. 下面关键字const有什么含意？

```
const int a;  
int const a;  
const int *a;  
int * const a;
```

(1)前两个的作用是一样，a是一个常整型数

(2)第三个意味着a是一个指向常整型数的指针（也就是，整型数是不可修改的，但指针可以）

(3)第四个意思a是一个指向整型数的常指针（也就是说，指针指向的整型数是可以修改的，但指针是不可修改的）

33.C语言中 extern的作用， extern c的作用？

(1)extern可以置于变量或者函数前，以标示变量或者函数的定义在别的文件中，提示编译器遇到此变量和函数时在其他模块中寻找其定义

(2)C++语言在编译的时候为了解决函数的多态问题，会将函数名和参数联合起来生成一个中间的函数名称，而C语言则不会，因此会造成链接时找不到对应函数的情况，此时C函数就需要用extern “C”进行链接指定，这告诉编译器，请保持我的名称，不要给我生成用于链接的中间函数名

34.obj-c是如何管理内存的？

OC语言使用引用计数来管理一个对象的生命周期，当对象被其他“人”使用的时候引用计数+1，不使用时-1，当此对象的引用计数为0时，则对象被系统回收。

35.什么是动态绑定(多态)?

动态绑定是面向对象特性之一多态的体现，声明的一个变量，在运行的时候可以绑定不同的对象，比如在方法的参数中声明这样一个变量UIView *view, 运行的时候，我们可以根据不同的应用场景给view变量传入不同的对象，可以传递一个UIButton对象，也可以传入UIImageView对象。

36.你是如何理解delegate

delegate，又称为 委托或代理，它是一种设计模式。delegate主要用于两个对象之间的通信交互，并且解除两个通信对象的耦合性，iOS中大量使用了delegate设计，主要用于视图与使用对象之间的通信交互。

37.block与函数有何异同，block有何优点

(1)block类似于C里面的函数指针，都可以作为参数进行传递，用于回调。但是block的实现可以定义在方法中，函数则不可以。

(2)block语法简洁，可以在方法中定义实现，这样可以访问方法中的局部变量，使代码更加的紧凑，结构化！

38.为什么标准头文件都有类似以下的结构？

```
#ifndef __INCvxWorksh
#define __INCvxWorksh
#ifdef __cplusplus
extern "C" {
#endif
/*...*/
#ifdef __cplusplus
}
#endif
#endif /* __INCvxWorksh */
```

显然，头文件中的编译宏“#ifndef __INCvxWorksh、#define __INCvxWorksh、#endif”的作用 是防止该头文件被重复引用。

39.cocoa touch框架

UIKit、Foundation、CoreGraphic、CoreAnimation

(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

40.栈结构与队列的区别？

栈（Stack）是限定只能在表的一端进行插入和删除操作的线性表。

队列（Queue）是限定只能在表的一端进行插入和在另一端进行删除操作的线性表。

1.队列先进先出，栈先进后出。

2. 对插入和删除操作的"限定"。栈是限定只能在表的一端进行插入和删除操作的线性表。队列是限定只能在表的一端进行插入和在另一端进行删除操作的线性表。
3. 遍历数据速度不同。队列遍历数据的速度要快得多。

41. OC中有无多重继承，如何实现多重继承

- (1) 没有多重继承。
- (2) cocoa中所有的类都是NSObject的子类，多重继承在这里是用protocol委托代理来实现的
- (3) 面向对象语言大部分都是单继承，如：java、c#
- (4) c++是支持多重继承的

42. static全局变量与普通的区别，局部变量呢？函数呢？

- (1) static全局变量与普通的全局变量有什么区别：static全局变量只初始化一次，防止在其他文件单元中被引用；
- (2) static局部变量和普通局部变量有什么区别：static局部变量只被初始化一次，下一次依据上一次结果值；
- (3) static函数与普通函数有什么区别：static函数在内存中只有一份，普通函数在每个被调用中维持一份拷贝

42. static,volatile区别，含义？

参考：http://bbs.ednchina.com/BLOG_ARTICLE_115375.HTM

43. 什么是类目与延展？

- (1) 类目：为已知的类增加新的方法
- (2) 延展：通知在本类的定义里使用类目来声明私有方法

44. OC中的协议和java中的接口区别？

- (1) 协议定义的方法，可以设置可选实现、必须实现，java中的接口则必须全都实现

45. 在一个对象的方法里面: self.name = "object"; 和 name = "object" 有什么不同吗？

前者调用的是设置器方法，后者是普通赋值

46.单例设计模式的实现，为什么使用单例设计

```
static File *instance = nil;

@implementation File

//获取单例的方法
+ (id)shareInstance {
    @synchronized (self) {
        if (instance == nil) {
            instance = [[File alloc] init];
        }
    }
    return instance;
}

//覆写allocWithZone、copyWithZone、retain、release、
autorelease 方法，目的是限制这个类只创建一个对象
```

为什么使用单例设计？

- 1.单例设计是用来限制一个类只能创建一个对象，那么此对象中的属性可以存储全局共享的数据，所有的类都可以访问、设置此单例对象中的属性数据
- 2.如果一个类创建的时候非常的耗费性能，那么此类可以设置为单例节约性能，如果能满足需求。

47.id类型、nil类型是什么？id与void * 是一样吗？

48.

49.

50.

iPhone UI笔试

1.层CALayer和UIView的区别是什么

两者最大的区别是,图层不会直接渲染到屏幕上，UIView是iOS系统中界面元素的基础，所有的界面元素都是继承自它。它本身完全是由CoreAnimation来实现的。它真正的绘图部分，是由一个CALayer类来管理。UIView本身更像是一个CALayer的管理器。一个UIView上可以有n个CALayer，每个layer显示一种东西，增强UIView的展现能力。

2.什么是GCD

GCD是Apple开发的一个多核编程的较新的解决方法。在Mac OS X 10.6雪豹中首次推出，并在最近引入到了iOS4.0。GCD是一个替代诸如NSThread等技术的很高效和强大的技术。GCD完全可以处理诸如数据锁定和资源泄漏等复杂的异步编程问题。

3.TCP/UDP区别联系

- (1)TCP的全称为传输控制协议。这种协议可以提供面向连接的、可靠的、点到点的通信。
- (2)UDP的全称为用户数据报协议，它可以提供非连接的不可靠的点到多点的通信。
- (3)用TCP还是UDP，那要看你的程序注重哪一个方面？可靠还是快速？

4.socket连接和http连接的区别

- (1)http连接：http连接就是所谓的短连接，即客户端向服务器端发送一次请求，服务器端响应后连接即会断掉；
- (2) socket连接：socket连接就是所谓的长连接，理论上客户端和服务器端一旦建立起连接将不会主动断掉

5.什么是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：客户端收到服务器的SYN + ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

6.frame与bounds的区别

- (1) Frame指的是：该view在父view坐标系统中的位置和大小（参照点是父视图的坐标系）
- (2) Bounds指的是：该view在本身坐标系统中的位置和大小（参照点是本身的坐标系）

7.如何理解MVC设计模式

MVC是一种架构设计，M表示数据模型、V表示视图、C表示控制器。

- (1)数据模型：负责存储、定义、操作数据
- (2)视图：用来展示数据给用户，和用户进行操作交互
- (3)控制器是M与V的协调者，控制器获取数据，将数据交给视图去展示

8.控制器的loadView方法是什么时候调用的，viewWillAppear什么时候调用的

- (1) loadView调用：当控制器的根视图view为空，且此view被访问时则调用
- (2) viewWillAppear：当控制器根视图view被添加到父视图上时调用

9.控制器的xib是怎么加载的

当UIViewController的loadView被调用时，在此方法中，通过NSBundle加载xib，先通过控制器的类名作为xib的文件名加载此xib文件。如果找到此xib文件，则加载为view对象作为控制器的根视图，如果没有xib文件，则通过alloc创建一个view对象作为根视图。

10.内存不足，系统会发出警告，此时控制器应该如何处理

内存不足时，系统会调用控制器的 didReceiveMemoryWarning 方法通知控制器内存不足。iOS6.0与6.0之前的处理方式不一样。

(1)6.0之前，调用 didReceiveMemoryWarning后，将self.view设置为nil,并且再调用viewDidUnload方法，在此方法中我们应该释放子视图！

(2)6.0之后，调用.didReceiveMemoryWarning后，不再调用viewDidUnload方法，则应该在 didReceiveMemoryWarning方法中手动将self.view=nil，并且释放子视图！

11.如何监听View的触摸事件，事件是如何传递的、视图的响应者链是什么

(1)覆写View类的touchBegin、touchMove、touchEnd系列方法监听视图的触摸。

(2)事件传递：当触摸一个视图时，首先系统会捕捉此事件，并为此事件创建一个UIEvent对象，将此对象加入当前应用程序的事件队列中，然后由UIApplication对象从队列中，一个一个取出来进行分发，首先分发给UIWindow对象，然后由UIWindow对象分发给触摸的视图对象，也就是第一响应者对象。

(3)响应者链：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它处理，如果它不处理，事件就会被传递给它的视图控制器对象（如果存在），然后是它的父视图（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。但有时候可以在视图的响应方法中根据一些条件判断来决定是否需要继续传递事件。



12. iOS中对象之间有哪些通信方式

代理delegate、block、通知、KVO

13.block在内存管理上的特点、需要注意循环引用，如何解决循环引用

(1)block块中使用了局部对象，则会将此对象retain，引用了当前对象的属性或者方法，则会将当前对象retain

(2)解决循环引用：将当前对象赋给一个局部变量，并且使用__block关键字修饰该局部变量，使用该变量访问当前对象的属性和方法

15.OC中有线程池(线程队列)吗，NSOperationQueue是如何管理线程的。

(1)NSOperationQueue是OC中的线程池，线程池中可以存放多个线程

(2)NSOperationQueue可以管理线程的并发数，还可以管理线程之间的优先级

16.timer的间隔周期准吗？为什么？怎样实现一个精准的timer？

定时器time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，这样就有可能导致定时器time会延迟从而不准确。我们可以开一个多线程，在此多线程上运行定时器，这样多线程只运行定时器，不会因堵塞而导致误差。

17.Http协议的特点，GET请求与POST请求的区别, 什么是https协议

(1)http协议是短链接，都是客户端主动发送请求，服务器做出响应，服务器响应之后，链接则断开。GET请求没有请求体，POST含有请求体，请求参数可以放入请求体中，所以POST可以提交大量的数据给服务器。

(2)什么是HTTPS：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，用于在客户计算机和服务器之间交换信息。它使用安全套接字层(SSL)进行信息交换，简单来说它是HTTP的安全版

```
Cup *cup = [[Cup alloc] init];
@try {
    //有可能出错的代码，例如数组越界
    [cup fill];
} @catch (NSException *exception) {
    //出错后执行的代码，没出错不执行
    NSLog(@"main: Caught %@: %@", [exception name],
        [exception reason]);
} @finally {
    [cup release];
}
```

18.xml数据的解析方式，各有什么不同， json解析有哪些框架

(1)xml数据解析有两种解析方式：DOM解析与SAX解析

(2)DOM解析必须先完成DOM树的构造，在处理规模较大的XML文档时就很耗内存，占用资源较多

(3)与DOM不同的是，它是用事件驱动模型,解析XML文档时每遇到一个开始或者结束标签、或者属性、或者一条指令时，程序就产生一个事件来进行相应的处理,因此，SAX相对于DOM来说更适合操作大的XML文档

(4)json解析现在性能比较好的主要是第三方的JSONKIT和iOS自带的json解析类，自带的json解析性能是最高的，但是只能是iOS5之后才可以使用

19.用过哪些开源网络框架，各有什么特点

(1)ASIHTTPRequest

优点：出来比较早、功能强大、文档丰富（目前很多应用还在使用）

缺点：已经停止更新、新特性少、厚重且对ios5以上支持不够完善。

(2)AFNetworking

优点：支持比较新的特性、简单易用

缺点：文档数目一般、功能少、忽略了一些扩展功能(如：没有同步请求)

(3)MKNetworkKit（建议使用）

优点：支持ARC、一个印度大卡写的。有ASIHTTPRequest的功能，AFNetworking的轻便

缺点：文档数目最少（几乎没有）

20.iOS中有哪些数据持久化的方式，各有什么特点，iOS平台怎么做数据的持久化？CoreData和sqlite有无必然联系？

CoreData是一个关系型数据库吗？

(1)主要有四种持久化方式：属性列表、对象归档、SQLite数据库、CoreData

(2)CoreData不是一个数据库，不过可以使用SQLite数据库来保持数据，也可以使用其他方式来存储数据，例如：XML

(3)属性列表、对象归档适合小数据量存储和查询操作

(4)SQLite、CoreData适合大数据量存储和查询操作

21.iPhone5如何适配

(1)添加启动图片，Default-568h@2x.png 分辨率640 * 1136 pixels

(2)在编码时，尽量不要写死480x320的尺寸，使用UIScreen获取硬件的物理尺寸

(3)通过设置 autosizing 来适应在父视图中的frame

22.iOS中如何捕捉异常

@try{

@catch{

23.用过单元测试吗

iOS自带了一个单元测试框架 OUnit，但目前最好用的测试框架应该是 GHUnit

24.OC有多继承吗，有什么替代方案

OC语言没有多继承，可以通过实现协议替代。OC是单继承，所有的类全都继承于NSObject。大部分面向对象语言都是单继承，C++是多继承

25.苹果APP的上架流程、什么情况下会被打回

- (1)访问私有API
 - (2)APP有严重的BUG
 - (3)给苹果审核的测试账号无法登陆，或者境外无法访问国内服务器
 - (4)APP描述中带“Beta”字样，或是其他表明APP还未开发完成的信息
 - (5) APP加载时间过长,iOS APP的最长启动时间不得超过15秒
 - (6) 给出外部购买链接
 - (7) APP描述中提到了iOS之外的其他支持平台
-

26.tableView是如何复用的

如果屏幕上能显示10个单元格，则tableView只会创建11个单元格，也就是n+1，当滑到第12个单元格时就会复用第1个单元格对象。tableView中有个单元格池的这么一个概念，tableView调用协议方法获取单元格时，先从池子中查找是否有可复用的单元格，如果有则复用，如果没有则创建一个单元格对象

27.如何优化tableView的滑动速度

- (1)复用单元格
- (2)使用不透明的视图，单元格中尽量少使用动画
- (3)图片加载使用异步加载，并且设置图片加载的并发数
- (4)滑动时不加载图片，停止滑动开始加载
- (5)文字、图片可以直接drawInRect绘制
- (6)如非必要，减少 reloadData 全部 cell，只 reloadDataAtIndexPaths
- (7)如果 cell 是动态行高，计算出高度后缓存
- (8)cell 高度固定的话直接用 cell.rowHeight 设置高度

28.ASI网络框架有哪些功能

- (1)ASI是款极其强劲的HTTP访问开源框架。
- (2)功能：异步请求、队列请求、GZIP压缩、缓存、断点续传、进度跟踪、上传文件、HTTP认证

29.了解Socket的通信方式吗

Socket的通信是通过TCP/IP协议，实现客户端与服务器端之间的通信方式。客户端通过3次握手与服务器建立可靠的连接，然后进行数据传输。

30.objective-c中对象之间是如何通信(交互)的？

对象之间的通信可以通过代理delegate、block、通知、KVO

31.如何将产品进行多语言发布，做国际化开发

- (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)在不同语言的 Localizable.strings文件中添加对应的文本
- (3)XIB文件国际化
- (4)程序名称国际化

参考：<http://hi.baidu.com/yunhuaikong/item/d8e8e2e4be13a8088c3ea829>

32.控制器loadView, viewDidLoad, viewDidUnload、didReceiveMemoryWarning 分别是在什么时候调用的

- (1)loadView: 根视图view为空时，且view被访问时调用
- (2) viewDidLoad: loadView调用完之后调用
- (3) viewDidUnload: iOS6.0之前，内存警告时调用
- (4) didReceiveMemoryWarning: 内存警告时调用

33.内存警告，iOS6与iOS6之前的版本处理方式的区别

- (1)iOS6之前是在 viewDidUnload中，释放子视图
- (2)iOS6之后在 didReceiveMemoryWarning 方法中释放子视图

34.谈谈对RunLoop的理解

Run loops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

35.如何调用drawRect方法与layoutSubviews方法，这两个方法的作用？

- (1)通过setNeedsLayout方法异步调用layoutSubviews方法
- (2)通过setNeedsDisplay方法异步调用drawRect方法
- (3)drawRect方法用于绘图，layoutSubviews方法用于布局子视图

36.UIView与UIViewController的关系？

每个控制器都有一个根视图，控制器UIViewController用来管理此视图的加载与显示的。

37.iOS中都有哪些手势

iOS提供了很多手势对象用于识别不同的手势操作，手势：轻击、捏合、平移、轻扫、旋转、长按

38.代理、通知和KVO的区别以及它们的优缺点

代理是一种回调机制，且是一对一的关系，而通知是一对多的关系，一个中心对象向所有的观察者提供变更通知（KVO是被观察者向观察者直接发送通知，这是通知和KVO的区别）。

39.http和socket区别

http是客户端用http协议进行请求，发送请求时候需要封装http请求头，并绑定请求的数据，服务器一般有web服务器配合（当然也非绝对）。http请求方式为客户端主动发起请求，服

务器才能给响应，一次请求完毕后则断开连接，以节省资源。服务器不能主动给客户端响应（除非采取http长连接技术）。iphone主要使用类是NSURLConnection。

socket是客户端跟服务器直接使用socket“套接字”进行连接，并没有规定连接后断开，所以客户端和服务端可以保持连接通道，双方都可以主动发送数据。一般在游戏开发或股票开发这种要求即时性很强并且保持发送数据量比较大的场合使用。主要使用类是CFSocketRef。

40. 二维码扫描用过哪些类库，这些类库有什么特点？

二维码生成：QRGener

二维码扫描：zxing、ZBarSDK

二维码扫描，推荐使用zxing, 因为zxing可以自定义扫描区域，而且是开源的，但是这个库集成起来比较复杂

41. iOS5有哪些新特性？

(1)iCloud、定制UI、Storyboard、ARC、CoreImage滤镜、新增JSON解析类

42. iOS6有哪些新特性？

(1)UIRefreshControl水滴效果下拉刷新

(2)UICollectionView控件的使用

(3)SLComposeViewController新浪微博控件

(4)PassKit中Passbook

(5)AutoLayout自动布局

43. iOS7有哪些新的特性？

系统：

(1)全新的UI设计与交互

(2)Control Center控制中心

(3)全新的多任务Multitasking

(4)AirDrop共享文件

(5)全新的Safari浏览器交互体验

SDK：

(1)动态UIKit, 新增了UIDynamicItem委托，用来为UIView制定动态行为

(2)新增游戏引擎Sprite Kit Framework框架，类似于cocos2d

参考：<http://www.leiphone.com/ios7-sdk.html>

44. Storyboard用过吗，有什么特点？

(1)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xib文件，和他们之间的跳转交互

(2)优点：不用在为每个控制器创建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

(3)缺点：Storyboard单个文件，不利于团队协作开发

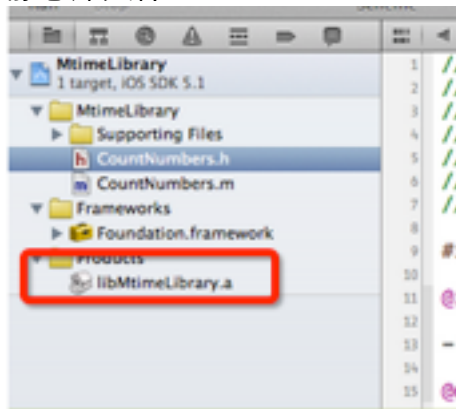
44. 用过ARC吗？ARC有什么特点

(1)ARC是编译器特性，iOS5.0新添加的特性，使用ARC，开发者不需要再retain、release、autorelease,因为编译器会在合适的地方自动插入retain,release

(2)特点：ARC不会因少release而导致内存泄露，过渡release导致程序崩溃
ARC可以产生更简洁的代码和更健壮的应用

45. 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。



46.APP发布的上架流程？

- (1)在苹果网站的开发者中心，新建一个APP，填写此APP相关的一些描述信息
- (2)下载安装发布证书
- (3)选择发布证书，使用Archive编译发布包
- (4)使用xcode提交发布包

47.SVN、Git协作开发，怎么防止代码文件冲突？

- (1)防止代码冲突：不要多人同时修改同一个文件。例如A、B都修改同一个文件，先让A修改，然后提交到服务器，然后B更新下来，再进行修改。
- (2)服务器上的项目文件 xcodeproj, 仅让一个人管理提交，其他人只更新。防止此文件产生冲突。

48.UIScrollerView用到了什么设计模式

49.断点续传是如何实现的？

50.你工作当中遇到最大问题是什么？

高级开发工程师

1.你使用过Objective-C的运行时编程（Runtime Programming）么？如果使用过，你用它做了什么？你还能记得你所使用的相关的头文件或者某些方法的名称吗？

2.你实现过多线程的Core Data么？NSPersistentStoreCoordinator，NSManagedObjectContext和NSManagedObject中的哪些需要在线程中创建或者传递？你是用什么样的策略来实现的？

- 3.Core开头的系列的内容。是否使用过CoreAnimation和CoreGraphics。UI框架和CA，CG框架的联系是什么？分别用CA和CG做过些什么动画或者图像上的内容。（有需要的话还可以涉及Quartz的一些内容）
- 4.是否使用过CoreText或者CoreImage等？如果使用过，请谈谈你使用CoreText或者CoreImage的体验。
- 5.NSNotification和KVO的区别和用法是什么？什么时候应该使用通知，什么时候应该使用KVO，它们的实现上有什么区别吗？如果用protocol和delegate（或者delegate的Array）来实现类似的功能可能吗？如果可能，会有什么潜在的问题？如果不能，为什么？（虽然protocol和delegate这种东西面试已经面烂了...）
- 6.你用过NSOperationQueue么？如果用过或者了解的话，你为什么要使用NSOperationQueue，实现了什么？请描述它和GCD的区别和类似的地方（提示：可以从两者的实现机制和适用范围来描述）。
- 7.既然提到GCD，那么问一下在使用GCD以及block时要注意些什么？它们两是一回事儿么？block在ARC中和传统的MRC中的行为和用法有没有什么区别，需要注意些什么？
- 8.您是否做过异步的网络处理和通讯方面的工作？如果有，能具体介绍一些实现策略么？
- 9.对于Objective-C，你认为它最大的优点和最大的不足是什么？对于不足之处，现在有没有可用的方法绕过这些不足来实现需求。如果可以的话，你有没有考虑或者实践过重新实现OC的一些功能，如果有，具体会如何做？
- 10.你实现过一个框架或者库以供别人使用么？如果有，请谈一谈构建框架或者库时候的经验；如果没有，请设想和设计框架的public的API，并指出大概需要如何做、需要注意一些什么方面，来使别人容易地使用你的框架。