1、objective-c 是所有对象间的交互是如何实现的？

在对象间交互中每个对象承担的角色不同，但总的来说无非就是”数据的发送者”或”数据的接收者”两种角色，我们可以通过代理去进行通信，或者通过观察者消息模式，block，delegagte

通过代理协议”或者”通知中心”方式的实现能最大限度的降低两交互对象之间的耦合；

2、objective-c中是如何实现线程同步的?

Mutexlock （互斥锁）、NSCondition lock （条件锁）消息传送

4、什么是UDP和TCP的区别是什么？

TCP 的全称是传输控制协议(Transmission Control Protocol)，这种协议可以提供面向连接的、可靠的、点到点的通信。

UDP 的全称是用户数据包协议(User Datagram Protocol)。他可以提供非连接的不可靠的点懂啊多点的通信，是OSI(Open System Interconnection的缩写，意为开放式系统互联。)参考模型中一种无连接的传输层协议，提供面向事务的简单的不可靠信息传输，\_IETF RFC 768 是UDP 的正式规范；

选择何种协议，看程序注重那个方面，可靠抑或快速。

5、TCP/IP 建立连接的过程？

在TCP/IP 协议中，TCP协议提供可靠的连接服务，采用三次握手建立连接；(所谓的三次握手，就是要有三次连接信息的发送、接收过程。TCP连的建立需要进行三次连接信息的发送、接收)

第一次握手：建立连接时，客户端发送连接请求到服务器，并进入SYN\_SEND状态，等待服务器确认；

第二次握手：服务器收到客户端连接请求，向客户端发送允许连接应答，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的允许连接应答，向服务器发送确认，客户端和服务器进入通信状态，完成三次握手。

9、编程中，保存数据有哪几种方式？

数组、自定义类、文件操作，数据库：如（SQLite 、core data、 plist 、NSUserDefaults、file ）用来持久保存数据

11、什么是抽象类？抽象类有什么作用？能实例化吗？

在c++中，含有纯虚拟函数的类称为抽象类，他不能生成对象；

在java中，含有抽象方法的类称为抽象类，同样不能生成对象；

抽象类是不完整的，没有完全实现，只能用作基类；不能实例化。

12、你对@interface和@property的理解？

@interface 声明一个类

@property就是声明2个函数即getter和setter，编译器会自动产生2个函数；

18、assign和retain ，copy区别？

assign就是和普通语言的赋值一样，不改变引用计数

retain 在一个对象中保存该对象需要拥有的对象，对该对象引用计数+1，指针拷贝；

copy 对NSString，内容拷贝，拷贝之前都会释放旧的对象

retain 对其他NSObject 和其子类

readonly 表示这个属性是只读的，就是只生成getter 方法，不生成setter 方法。

　　8.OC如何对内存管理的,你的看法和解决方法?

　　OC的内存管理主要有三种方式ARC(自动内存计数)、(MRC)手动内存计数、(NSAutoRealeasePool)内存池。

1. ARC是iOS 5推出的新功能，全称叫 ARC(Automatic Reference Counting)。简单地说，就是代码中自动加入了retain/release，原先需要手动添加的用来处理内存管理的引用计数的代码可以自动地由编译器完成了。

该机能在 iOS 5/ Mac OS X 10.7 开始导入，利用 Xcode4.2 可以使用该机能。简单地理解ARC，就是通过指定的语法，让编译器(LLVM 3.0)在编译代码时，自动生成实例的引用计数管理部分代码。有一点，ARC并不是GC，它只是一种代码静态分析（Static Analyzer）工具。

　　解决: 通过alloc – init方式创建的, 创建后引用计数+1, 此后每retain一次引用计数+1, 那么在程序中做相应次数的release就好了.

2. 谁创建，谁释放(谁污染，谁治理)

谁调用new,alloc,retain,copy,multableCopy,谁就要调用对应的release或autorelease;

　　解决:一般是由类的静态方法创建的, 函数名中不会出现alloc或init字样, 如[NSString string]和[NSArray arrayWithObject:], 创建后引用计数+0, 在函数出栈后释放, 即相当于一个栈上的局部变量. 当然也可以通过retain延长对象的生存期.

　　3. (NSAutoRealeasePool)内存池：可以通过创建和释放内存池控制内存申请和回收的时机.

解决:是由autorelease加入系统内存池, 内存池是可以嵌套的, 每个内存池都需要有一个创建释放对, 就像main函数中写的一样. 使用也很简单, 比如[[[NSString alloc]initWithFormat:@”Hello world!”] autorelease], 即将一个NSString对象加入到最内层的系统内存池, 当我们释放这个内存池时, 其中的对象都会被释放.

　　13. 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:

　　14.描述一下iOS SDK中如何实现MVC的开发模式

　　MVC是模型、试图、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。

　　16. 类别的作用?继承和类别在实现中有何区别?

　1.类别中不能添加新的成员变量；

2.类别中不能添加新的属性，但能够生成setter getter方法的声明，但不能生成setter getter 方法的实现,也不能够生成带下划线的私有成员变量；

3.类别可以添加类方法和对象方法

4.类别中的方法优先级高于原有类的方法（类别中方法优先级更高）

5.开发中尽量不要使原有类的方法和类别方法重复　　类别主要有3个作用：

　　(1)将类的实现分散到多个不同文件或多个不同框架中。

　　(2)创建对私有方法的前向引用。

　　(3)向对象添加非正式协议。

　　继承可以增加，修改或者删除方法，并且可以增加属性。

　　18. oc中的协议和java中的接口概念有何不同?

　　答案：OC中的代理有2层含义，官方定义为 formal和informal protocol。前者和Java接口一样。

　　informal protocol中的方法属于设计模式考虑范畴，不是必须实现的，但是如果有实现，就会改变类的属性。

　　“非正式协议概念其实就是类别的另一种表达方式“这里有一些你可能希望实现的方法，你可以使用他们更好的完成工作”。

　　这个意思是，这些是可选的。比如我门要一个更好的方法，我们就会申明一个这样的类别去实现。然后你在后期可以直接使用这些更好的方法。

　　现在来看，其实protocal已经开始对两者都统一和规范起来操作，因为资料中说“非正式协议使用interface修饰“，

　　现在我们看到协议中两个修饰词：“必须实现(@requied)”和“可选实现(@optional)”。

　　21. oc中可修改和不可以修改类型。

　　比如NSArray和NSMutableArray。前者在初始化后的内存控件就是固定不可变的，后者可以添加等，可以动态申请新的内存空间。

　　22. 我们说的oc是动态运行时语言是什么意思?

　　答案：多态。 主要是将数据类型的确定由编译时，推迟到了运行时。

　　这个问题其实浅涉及到两个概念，运行时和多态。

　　简单来说，运行时机制使我们直到运行时才去决定一个对象的类别，以及调用该类别对象指定方法。

　　多态：不同对象以自己的方式响应相同的消息的能力叫做多态。意思就是假设生物类(life)都用有一个相同的方法-eat;

　　那猫属于生物，狗也属于生物，都继承了life后，实现各自的eat，但是调用是我们只需调用各自的eat方法。

　　也就是不同的对象以自己的方式响应了相同的消息(响应了eat这个选择器)。

　　因此也可以说，运行时机制是多态的基础

　　28. frame和bounds有什么不同?

　　答案:frame指的是：该view在父view坐标系统中的位置和大小。(参照点是父类的坐标系统)

　　bounds指的是：该view在本身坐标系统中 的位置和大小。(参照点是本身坐标系统)

　　29. 方法和选择器有何不同?

　　答案：selector是一个方法的名字，method是一个组合体，包含了名字和实现.

　　32.延迟加载(延时加载)：只在用到的时候才去初始化,tableView中图片的加载显示了。一个延时载，避免内存过高，一个异步加载，避免线程堵塞。

　　35. 什么时候使用NSMutableArray，什么时候使用NSArray?

　　答案：当数组在程序运行时，需要不断变化的，使用NSMutableArray，当数组在初始化后，便不再改变的，使用NSArray。需要指出的是，使用NSArray只表明的是该数组在运行时不发生改变，即不能往NSAarry的数组里新增和删除元素，但不表明其数组內的元素的内容不能发生改变。NSArray是线程安全的，NSMutableArray不是线程安全的，多线程使用到NSMutableArray需要注意。

　　36. 给出委托方法的实例，并且说出UITableVIew的DataSource方法

　　答案：CocoaTouch框架中用到了大量委托，其中UITableViewDelegate就是委托机制的典型应用，是一个典型的使用委托来实现适配器模式，其中UITableViewDelegate协议是目标，tableview是适配器，实现UITableViewDelegate协议，并将自身设置为talbeview的delegate的对象，是被适配器，一般情况下该对象是UITableViewController。

　　UITableVIew的DataSource方法有- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section;

　　- (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath;

　　40. 类NSObject的那些方法经常被使用?

　　答案：NSObject是Objetive-C的基类，其由NSObject类及一系列协议构成。

其中类方法alloc、class、 description

对象方法int、dealloc、– performSelector:withObject:afterDelay:等经常被使用

43. UIView的动画效果有那些?

UIViewAnimationOptionCurveEaseInOut UIViewAnimationOptionCurveEaseIn UIViewAnimationOptionCurveEaseOut UIViewAnimationOptionTransitionFlipFromLeft UIViewAnimationOptionTransitionFlipFromRight UIViewAnimationOptionTransitionCurlUp

UIViewAnimationOptionTransitionCurlDown

　　44. 在iPhone应用中如何保存数据?

　　答案：有以下几种保存机制：

　　1.通过web服务，保存在服务器上

　　2.通过NSCoder固化机制，将对象保存在文件中

　　3.通过SQlite或CoreData保存在文件数据库中

　　45. 什么是coredata?

　　答案：coredata是苹果提供一套数据保存框架，它基于SQlite

　　46. 什么是NSManagedObject模型?

　　答案：NSManagedObject是NSObject的子类 ，也是coredata的重要组成部分，它是一个通用的类,实现了core data 模型层所需的基本功能，用户可通过子类化NSManagedObject，建立自己的数据模型。

　　47. 什么是NSManagedobjectContext?

　　答案：NSManagedobjectContext对象负责应用和数据库之间的交互。

　　48. 什么是谓词?

　　答案：谓词是通过NSPredicate，是通过给定的逻辑条件作为约束条件，完成对数据的筛选。

　　predicate = [NSPredicate predicateWithFormat:@"customerID == %d",n];

　　a = [customers filteredArrayUsingPredicate:predicate];

　　50. 谈谈对Block 的理解?并写出一个使用Block执行UIVew动画?

　　答案：Block是可以获取其他函数局部变量的匿名函数，其不但方便开发，并且可以大幅提高应用的执行效率(多核心CPU可直接处理Block指令)

　　[UIView transitionWithView:self.view

　　duration:0.2

　　options:UIViewAnimationOptionTransitionFlipFromLeft

　　animations:^{ [[blueViewController view] removeFromSuperview]; [[self view] insertSubview:yellowViewController.view atIndex:0]; }

　　completion:NULL];

　　53. 做过的项目是否涉及网络访问功能，使用什么对象完成网络功能?

ASIHTTPRequest与NSURLConnection(NSURLSession)

　　56. iOS 中的多线程

　　iOS中的多线程，是Cocoa框架下的多线程，通过Cocoa的封装，可以让我们更为方便的使用线程，做过C++的同学可能会对线程有更多的理解，比如线程的创立，信号量、共享变量有认识，Cocoa框架下会方便很多，它对线程做了封装，有些封装，可以让我们创建的对象，本身便拥有线程，也就是线程的对象化抽象，从而减少我们的工程，提供程序的健壮性。

　　GCD是(Grand Central Dispatch)的缩写 ，从系统级别提供的一个易用地多线程类库，具有运行时的特点，能充分利用多核心硬件。GCD的API接口为C语言的函数，函数参数中多数有Block，关于Block的使用参看这里，为我们提供强大的“接口”，对于GCD的使用参见本文

　　NSOperation与Queue

　　NSOperation是一个抽象类，它封装了线程的细节实现，我们可以通过子类化该对象，加上NSQueue来同面向对象的思维，管理多线程程序。具体可参看这里：一个基于NSOperation的多线程网络访问的项目。

　　NSThread

　　NSThread是一个控制线程执行的对象，它不如NSOperation抽象，通过它我们可以方便的得到一个线程，并控制它。但NSThread的线程之间的并发控制，是需要我们自己来控制的，可以通过NSCondition实现。

　　参看 iOS多线程编程之NSThread的使用

　　其他多线程

　　在Cocoa的框架下，通知、Timer和异步函数等都有使用多线程……

　　57. 在项目什么时候选择使用GCD，什么时候选择NSOperation?

　　项目中使用NSOperation的优点是NSOperation是对线程的高度抽象，在项目中使用它，会使项目的程序结构更好，子类化NSOperation的设计思路，是具有面向对象的优点(复用、封装)，使得实现是多线程支持，而接口简单，建议在复杂项目中使用。

　　项目中使用GCD的优点是GCD本身非常简单、易用，对于不复杂的多线程操作，会节省代码量，而Block参数的使用，会是代码更为易读，建议在简单项目中使用。

　　58. 什么是block

　　对于闭包(block),有很多定义，其中闭包就是能够读取其它函数内部变量的函数，这个定义即接近本质又较好理解。对于刚接触Block的同学，会觉得有些绕，因为我们习惯写这样的程序main(){ funA();} funA(){funB();} funB(){…..}; 就是函数main调用函数A，函数A调用函数B… 函数们依次顺序执行，但现实中不全是这样的，例如项目经理M，手下有3个程序员A、B、C，当他给程序员A安排实现功能F1时，他并不等着A完成之后，再去安排B去实现F2，而是安排给A功能F1，B功能F2，C功能F3，然后可能去写技术文档，而当A遇到问题时，他会来找项目经理M，当B做完时，会通知M，这就是一个异步执行的例子。在这种情形下，Block便可大显身手，因为在项目经理M，给A安排工作时，同时会告诉A若果遇到困难，如何能找到他报告问题(例如打他手机号)，这就是项目经理M给A的一个回调接口，要回掉的操作，比如接到电话，百度查询后，返回网页内容给A，这就是一个Block，在M交待工作时，已经定义好，并且取得了F1的任务号(局部变量)，却是在当A遇到问题时，才调用执行，跨函数在项目经理M查询百度，获得结果后回调该block。

　　3 block的使用

　　使用实例

　　cocoaTouch框架下动画效果的Block的调用

　　使用typed声明block

　　typedef void(^didFinishBlock) (NSObject \*ob);

　　这就声明了一个didFinishBlock类型的block，

　　然后便可用

　　@property (nonatomic,copy) didFinishBlock finishBlock;

　　声明一个blokc对象，注意对象属性设置为copy，接到block 参数时，便会自动复制一份。

　　\_\_block是一种特殊类型，

　　使用该关键字声明的局部变量，可以被block所改变，并且其在原函数中的值会被改变。

　　4 常见系列面试题

　　面试时，面试官会先问一些，是否了解block，是否使用过block，这些问题相当于开场白，往往是下面一系列问题的开始，所以一定要如实根据自己的情况回答。

　　1 使用block和使用delegate完成委托模式有什么优点?

　　首先要了解什么是委托模式，委托模式在iOS中大量应用，其在设计模式中是适配器模式中的对象适配器，Objective-C中使用id类型指向一切对象，使委托模式更为简洁。了解委托模式的细节：

　　iOS设计模式—-委托模式

　　使用block实现委托模式，其优点是回调的block代码块定义在委托对象函数内部，使代码更为紧凑;

　　适配对象不再需要实现具体某个protocol，代码更为简洁。

　　2 多线程与block

　　GCD与Block

　　使用 dispatch\_async 系列方法，可以以指定的方式执行block

　　GCD编程实例

　　dispatch\_async的完整定义

　　void dispatch\_async(

　　dispatch\_queue\_t queue,

　　dispatch\_block\_t block);

　　功能：在指定的队列里提交一个异步执行的block，不阻塞当前线程

　　通过queue来控制block执行的线程。主线程执行前文定义的 finishBlock对象

dispatch\_async(dispatch\_get\_main\_queue(),^(void){finishBlock();});