#### 应用运行周期

* 应用有哪几种状态、进入前台、后台切换有哪些方式

#### 属性

* 属性的实质

@property = ivar(变量) + getter + setter;  
  利用class\_copyPropertyList 查看类的所有属性  
  利用class\_copyIvarList查看类的所有成员变量  
  利用class\_copyMethodList查看类的所有方法

每次增加一个属性,系统都会在ivar\_list中添加一个成员变量的描述

在method\_list中增加setter与getter方法的描述

在prop\_list 中增加一个属性的描述，计算该属性在对象中的偏移量

然后给出 setter与 getter 方法对应的实现

在 setter 方法中从偏移量的位置开始赋值

在 getter 方法中从偏移量开始取值

注：为了能够读取正确字节数,系统对偏移量的指针类型进行了类型强转

编译器会自动生成一套存取方法，用以访问给定类型中具有给定名称的变量

* 属性特质

原子性 (atomic, nonatomic)

读写权限 (readonly, readwrite)

内存管理语义 (assign, retain, strong, weak,copy)

方法名

* 属性retain、strong、weak之间的区别以及atomic、nonatomic之间的区别
* ARC下默认属性是什么

基本数据类型：atomic、readwrite、assign

对象：atomic、readwrite、strong

* 属性weak修饰关键字原理以及如何实现weak修饰的变量在对象销毁的时候变量被置成nil
* \_\_weak、\_\_block有什么区别？
* @synthesize

@synthesize的语义是如果你没有手动实现setter方法和getter方法，那么编译器会自动为你加上这两个方法, 编译器在编译阶段执行自动合成

* @dynamic

@dynamic告诉编译器,属性的setter与getter方法由用户自己实现

#### 代码管理

* 代码管理工具有哪些？原理？
* CocoaPods原理？私有库的实现原理？私有库如何实现内部资源的引用

#### 设计模式

* MVC、单例、工厂、代理设计模式
* 组件化之间如何解耦
* 桥接
* 使用那些第三方库，能简单介绍几种常用的几个库的原理吗？
* 热修复的原理（JSPtch、ReactNative）
* 常用控件继承关系？
* 多线程技术有哪些？
* 运行时、RunLoop、Autorelease原理
* HTTP的GET、POST请求的区别
* HTTPS的加密原理
* 状态码有了解过吗
* 请简单写出增删改查的SQL语句
* NSURLSession的优势与缺点
* 开发项目中见到或使用过的锁？各自之间的原理有了解过吗？
* SDWebImage以及AFNetWorking使用的锁有哪些效率怎么样？
* 使用atomic一定是线程安全的吗
* 数据库建表的时候索引有什么用？表一定要有主键吗？数据迁移？
* iOS持久化的如何选择？项目中使用的持久化方案？谈谈优点
* iOS设备唯一设备号的历史变迁
* HTTP发送POST表单理解？
* 对方释放的时机
* 线程保活
* 协议
* 分类、子类、父类加载顺序
* load、Initian
* 可变不可变的实质
* 简单谈谈推送、本地、远程
* iOS缓存理解？如何设计缓存？SDWebImage、YYCache
* ARC原理，引用计数原理、规则
* 使用过RAC吗？简单谈谈原理
* UI布局的方式有哪些各自有什么优点缺点？
* 倒计时如何实现？（NSTimer？）
* isa指针？
* UITableView的优化
* init私有化方式
* 如何防止崩溃？检测崩溃原理、dsym原理
* 事件响应链
* block的实质？block持有变量如何避免循环引用
* GCD的原理
* 加载图片的方式有哪些各自的优缺点
* cell重用原理
* 如何实现断点下载上传功能
* 浅拷贝与深拷贝的区别
* \_cmd、self、super、superClass

\_cmd 苹果官方的解释是这样的：The \_cmd variable is a hidden argument passed to every method that is the current selector.

self 表示调用当前方法的对象实例

super只是编译特性，并不是一个指针，仅仅是让方法调用者调用父类方法

superClass 获取方法调用者的父类

Person \*person = [Person new];

[person test];

- (void)test {

// 打印

[self class], [self superClass], [super class], [super superClass]

[self class]表示person对象调用class方法得到类名Person

[self superclass]调用superclass方法,得到父类类名NSObject

[super class]调用父类的class方法, person对象得到类名Person

[super superclass]调用父类的superclass方法,得到父类类名NSObject

}

* SEL、IMP理解

SEL是一种类型，含义是方法的编号

SEL selector = @selector(myName);

IMP是函数的真正实现地址

IMP imp = class\_getMethodImplement([seld class], selector);

* SEL和@selector的含义

可以理解 @selector就是取类对象方法的编号,他的行为基本可以等同C语言的中函数指针,只不过C语言中，可以把函数名直接赋给一个函数指针，而Object-C的类不能直接应用函数指针,这样只能做一个@selector语法来取.它的结果是一个SEL类型.这个类型本质是类对象方法的编号(函数地址)

* 内存几大区域
* SEL 类成员方法的指针
* 汇编的简单知识
* 启动优化知识点
* 设计循环滚动
* 判断屏幕上有多少cell
* 哈希表
* 加密算法了解
* KVO、KVC实现原理
* 审核被拒之后如何处理
* if else 、switch效率为题？
* 内存分析、Instrument使用
* 圆角实现最优方案
* 网络模型几层？每个层简单分析一下？
* 有序、无需集合原理
* 数组、字典持有对象、引用计数会增加吗
* 核心动画了解吗
* Socket建立连接的原理
* TCP/IP协议了解过吗？UDP呢？如何建立长连接？传递数据的可靠性？
* 技术难点？
* 并发、串行
* 二叉树、链表
* 封装控件
* 个人规划
* 算法
* UIScrollView实现原理
* 坐标系的理解
* GCD信号量说说使用过吗？
* static、const、#define
* 堆栈的理解？区别联系
* 面向对象的理解？
* iOS缺点、还有就是Swift了解过吗
* 分类、匿名分类
* @synthesize @dynamic
* 混编
* 多态、动态绑定
* NSNull、null、nil、NULL
* UIView、CALayer的关系
* JSON、XML解析
* UIWebView、WKWebView怎样与js交互
* 进程、与线程之间的关系
* 函数指针
* @代表的含义
* 1. +load +initialize
* 2. 深浅拷贝，数组http://www.jianshu.com/p/49f01cc49b66
* 3. @selector和方法
* 4. autolayout写约束的方法
* 5. 二分查找，冒泡，快速排序，字符串反转
* 6. iphone的帧率是多少 60帧
* 7.什么定时器是准时的？ NSTimer(不是)、CADisplay(不是)、GCD(是)
* 8. KVO,KVC
* 9. NSTimer不循环引用的方法 两种方法 1.target是第三方对象弱引用self 2.target是自己
* 10. 类的结构\*\*\*\*\*\*\*
* 11. 写个自定义的imagepicker
* 12. 图片轮播
* 13. 如何写一个安全的数组\*\*\*\*\*\*
* 14. NSSet和NSArray的区别\*\*\*\*
* 15.HTTPS\*\*\*\*\*\*\*\*\*\*
* 16.TCP断开四次握手\*\*\*\*\*\*
* 17.有什么好用的开源库
* 18.有什么擅长的东西\*\*\*\*
* 19.在公司的App具体负责哪个模块
* 20.看过什么书\*\*\*\*
* 21. react native\*\*\*\*\*
* 22.jspatch有没有看过\*\*\*\*\*
* 23.404 200 304等等\*\*\*\*\*
* 24.keep-alive content length\*\*\*\*
* 26.事件响应者链\*\*\*\*\*
* 27.weak的底层实现\*\*\*\*\*
* 28.讲讲直播的原理\*
* 29.NSSet的原理
* 22.runtime如何通过selector找到对应的IMP地址？（分别考虑类方法和实例方法）
* 23 应用删除后如何保留信息钥匙串

keychain https://www.cnblogs.com/v2m\_/archive/2012/01/18/2325782.html

SecItemCopyMatching Returns one or more items which match a search query.

剪切版

* app之间如何传递数据
* 类、类对象、元类之间的关系(消息分发机制、消息机制、类的本质)

OC中的对象是一个指向类对象地址的变量

对象的实例变量实际上是距存放对象内存区域一定偏移量的内存区域

无论是类方法，还是对象方法， 调用者都必须是对象，就是说类其实也是一种对象

* 实例变量的本质