# 面试题笔记

1. 用obj-c写一个插入法排序,或你所知道的排序

插入法排序:<http://www.cnblogs.com/rosylxf/archive/2012/07/31/2617565.html>

冒泡排序:

-（NSArray\*）myBubble\_Sort:(NSArray\*)oldArray

{

        NSMutableArray \* newArray = [NSMutableArray arrayWithArray: oldArray];

        NSInterger num = [oldArray count];

        for(int i = 0 ; i < num-1 ; i++)

       {

            for（int j = i +1; j < num ; j++）

            {

                int num1 = [[oldArray objectAtIndex: i] intValue];

               int num2 = [[oldArray objectAtIndex: j] intValue];

              if(num1 < num2)

             {

                    [newArray replaceObjectAtIndex: i  withObject:[NSString stringWithFormat:@"%d",num2]];

                    [newArray replaceObjectAtIndex: j  withObject:[NSString stringWithFormat:@"%d",num1]];

            }

      }

    return newArray；

}

void bubble\_Sort(int a[], int n)

{

        int i , j ;

        for(i = 0 ; i < n-1 ; i++)

        {

                for(j = i+1; j < n; j++)

                    if(a[i] < a[j])

                    {

                               a[i] = a[i] + a[j];

                                a[j] = a[i] - a[j];

                                a[i] = a[i] - a[j];

                   }

        }

}

1. 简要描述如何提高iOS应用性能?

我要给出的建议将分为三个不同的等级： 入门级、 中级和进阶级：

入门级（这是些你一定会经常用在你app开发中的建议）

* 1. 用ARC管理内存
* 2. 在正确的地方使用reuseIdentifier
* 3. 尽可能使Views透明
* 4. 避免庞大的XIB
* 5. 不要block主线程
* 6. 在Image Views中调整图片大小
* 7. 选择正确的Collection
* 8. 打开gzip压缩

中级（这些是你可能在一些相对复杂情况下可能用到的）

* 9. 重用和延迟加载Views
* 10. Cache, Cache, 还是Cache！
* 11. 权衡渲染方法
* 12. 处理内存警告
* 13. 重用大开销的对象
* 14. 使用Sprite Sheets
* 15. 避免反复处理数据
* 16. 选择正确的数据格式
* 17. 正确地设定Background Images
* 18. 减少使用Web特性
* 19. 设定Shadow Path
* 20. 优化你的Table View
* 21. 选择正确的数据存储选项

进阶级（这些建议只应该在你确信他们可以解决问题和得心应手的情况下采用）

* 22. 加速启动时间
* 23. 使用Autorelease Pool
* 24. 选择是否缓存图片

25. 尽量避免日期格式转换

1. 什么是KVO和KVC

KVC是键值编码，可以通过一个字符串的key（属性名）修改对象的属性值.

KVO是键值监听，可以监听一个对象属性值的改变

1. 对于单例的理解

单例模式是一种常用的软件设计模式。在它的核心结构中只包含一个被称为单例类的特殊类。通过单例模式可以保证系统中一个类只有一个实例而且该实例易于外界访问，从而方便对实例个数的控制并节约系统资源。如果希望在系统中某个类的对象只能存在一个，单例模式是最好的解决方案

1. OC中的私有方法和私有变量

OC没有类似@private的修饰词来修饰方法，只要写在.h文件中，就是公共方法

可以直接在.m文件中（比如类扩展）声明和实现方法，对编译器来说是私有的

1. 控制器的生命周期

当一个视图控制器被创建，并在屏幕上显示的时候。 代码的执行顺序

1、 alloc                                   创建对象，分配空间

2、init (initWithNibName) 初始化对象，初始化数据

3、loadView                          从nib载入视图 ，通常这一步不需要去干涉。除非你没有使用xib文件创建视图

4、viewDidLoad                   载入完成，可以进行自定义数据以及动态创建其他控件

5、viewWillAppear              视图将出现在屏幕之前，马上这个视图就会被展现在屏幕上了

6、viewDidAppear               视图已在屏幕上渲染完成

当一个视图被移除屏幕并且销毁的时候的执行顺序，这个顺序差不多和上面的相反

1、viewWillDisappear            视图将被从屏幕上移除之前执行

2、viewDidDisappear             视图已经被从屏幕上移除，用户看不到这个视图了

3、dealloc                                 视图被销毁，此处需要对你在init和viewDidLoad中创建的对象进行释放

关于viewDidUnload ：在发生内存警告的时候如果本视图不是当前屏幕上正在显示的视图的话， viewDidUnload将会被执行，本视图的所有子视图将被销毁，以释放内存,此时开发者需要手动对viewLoad、viewDidLoad中创建的对象释放内存。 因为当这个视图再次显示在屏幕上的时候，viewLoad、viewDidLoad 再次被调用，以便再次构造视图。

1. 多线程与线程安全

线程安全:当调用线程A 让线程B停用的时候 发生一些元素丢失就是不安全的线程,反之就是线程安全.

1. 常见的object-c的数据类型有那些， 和C的基本数据类型有什么区别？

常用OC类型：NSString、NSArray、NSDictionary、NSData、NSNumber等

OC对象需要手动管理内存，C的基本数据类型不需要管理内存

1. id声明的变量有什么特性？

id声明的变量能指向任何OC对象

1. Objective-C如何对内存管理的,说说你的看法和解决方法?
2. 每个对象都有一个引用计数器，每个新对象的计数器是1，当对象的计数器 减为0时，就会被销毁
3. 通过retain可以让对象的计数器+1、release可以让对象的计数器-1
4. 还可以通过autorelease pool管理内存
5. 如果用ARC，编译器会自动生成管理内存的代码
6. 内存管理的几条原则时什么？按照默认法则.哪些方法生成的对象需要手动释放？在和property结合的时候怎样有效的避免内存泄露？
7. 只要调用了alloc、copy、new方法产生了一个新对象，都必须在最后调用一次release或者autorelease
8. 只要调用了retain，都必须在最后调用一次release或者autorelease
9. @property如果用了copy或者retian，就需要对不再使用的属性做一次release操作
10. 如果用了ARC，另外讨论
11. 什么是代理
12. 两个对象之间的传递数据和消息
13. 解耦,拆分业务逻辑
14. 什么时候用NSArray 什么时候用NSMutableArray?
15. 当数组元素需要动态的添加或者删除时候用NSMutableArray
16. 当数组元素固定不变的时候用NSArray
17. aotorelease自动释放池有限制么?

没有限制

1. 什么是深拷贝什么是浅拷贝?

浅拷贝：指针（地址）拷贝，不会产生新对象

深拷贝：内容拷贝，会产生新对象

1. Object-C有多继承吗?没有的话用什么代替?

OC是单继承,没有多继承

有时候可以用分类和协议来代替多继承

17. 线程和进程的区别？

1. 一个应用程序对应一个进程，一个进程帮助程序占据一块存储空间
2. 要想在进程中执行任务，就必须开启线程，一条线程就代表一个任务
3. 一个进程中允许开启多条线程，也就是同时执行多个任务

18. 堆和栈的区别？

1. 堆空间的内存是动态分配的，一般存放对象，并且需要手动释放内存
2. 栈空间的内存由系统自动分配，一般存放局部变量等，不需要手动管理内存

19. 为什么很多内置的类，如TableView的delegate的属性是assign不是retain？

1. tableView的代理一般都是它所属的控制器，控制器会对它内部的view做一次retain操作
2. 假设tableView也对代理（控制器）做一次retain操作，那么就出现循环retain问题

20. 定义属性时，什么情况使用copy、assign、retain？

1. copy：NSString、Block等类型
2. assign：基本数据类型
3. retain：OC对象类型

21. tableView的重用机制？

这里只是简述：将离开屏幕的cell放到缓存池，重新拿来显示到屏幕的其他位置（其他自己详细描述）

22. 怎么理解MVC，在Cocoa中MVC是怎么实现的？

1. M：Model，模型，封装数据
2. V：View，视图界面，负责展示数据
3. C：Controller，控制器，负责提供数据（Model）给界面（View）

23. 你用的是什么推送,自己会写吗?

1> 极光推送,友盟推送

2> 自己写的

24. 项目上遇到过什么难题吗?

25. 项目中遇到难题你是怎么解决的?

26. 什么是Block

1. Block是OC中的一种数据类型

2.是一个能工作的代码单元，可以在任何需要的时候被执行

3.本质上是轻量级的匿名函数，可以作为其他函数的参数或者返回值

4.块代码本身可能有一个参数列表，也可能有一个返回值

5.可以把块代码赋给一个变量，并在需要的时候调用，就像调用一个普通函数

27. 瀑布流怎么实现的

提示:UICollectionView

28. 去年的WWDC 你看了吗? 有什么新API和新技术公布

提示: Touch ID技术 Swift语言 Watch Kit

29. OBJ-C的优缺点

objc优点：

  1) Cateogies

  2) Posing

  3) 动态识别

  4) 指标计算

  5）弹性讯息传递

  6) 不是一个过度复杂的 C 衍生语言

  7) Objective-C 与 C++ 可混合编程

缺点:

  1) 不支援命名空间

  2)  不支持运算符重载

  3）不支持多重继承

  4）使用动态运行时类型，所有的方法都是函数调用，所以很多编译时优化方法都用不到。（如内联函数等），性能低劣。

30. 本地缓存的原理

提示: 私人通讯录

人事面试题

31. 请自我介绍一下自己

32. 你觉得你的优点是啥

33. 说说你的缺点

34. 你对加班的看法

35. 你对薪资的要求

36. 你以后的职业规划是什么

37. 当工作遇到问题,你会怎么办

38. 你还有什么要问的吗?

39. 你喜欢什么系的菜

40. 你对我们公司有什么看法

41. 你为什么会投简历到我们公司

42. 你以前知道我们公司吗

43. 你为什么要离职

44. 谈谈你对好员工的看法

45. 谈谈你工作习惯

46. 想过创业吗

47. 怎么样看待学历和能力

48. 你今年的目标是什么

49. 你爱玩游戏吗? 会不会因为游戏耽搁工作

50. 你对公司文化的了解是什么