今天在东软熙康健康科技有限公司  
的面试题  
  
1、Object-C有多继承吗？没有的话用什么代替？  
没，protocol.  
2、Object-C有私有方法吗？私有变量呢？  
没，有。  
3、#import和#include的区别，@class代表什么？  
#import 能防止重复引用，#include 不能，@class 前置声明一个类。  
4、Object-C的内存管理？  
引用计数器。  
5、为什么很多内置的类，如TableViewController的delegate的属性是assign不是retain？  
防止循环引用。  
6、定义属性时，什么情况使用copy、assign、retain？  
copy:拷贝对象。assign:传值，一般用于基础类型。retain:传地址，引用计数器加一。  
7、对象是什么时候被release的？  
引用计数器为0.  
8、iOS有没有垃圾回收？  
没。  
9、tableView的重用机制？  
复用标记。  
10、ViewController 的loadView、viewDidLoad、viewDidUnload分别是什么时候调用的，在自定义ViewController时在这几个函数中应该做什么工作？  
当view为nil时调用loadView,view完成加载调用viewDidLoad，view释放时调用viewDidUnload.  
11、ViewController的didReceiveMemoryWarning是在什么时候调用的？默认的操作是什么？  
内存超过阙值，尝试释放view.  
12、列举Cocoa中常见的集中多线程的实现，并谈谈多线程安全的几种解决办法，一般什么地方会用到多线程？  
NSThread,GCD等。尽量用上层分装好的方法去实现多线程而不是手动调用NSThread。  
13、怎么理解MVC，在Cocoa中MVC是怎么实现的？  
Model——View——Controller。  
14、delegate和notification区别，分别在什么情况下使用？  
  
15、self.跟self什么区别？  
16、id、nil代表什么？  
id相当于void\*，nil是空对象。  
17、内存管理 Autorelease、retain、copy、assign的set方法和含义？  
18、类别的作用？  
19、委托（举例）  
20、retainCount？  
  
21、ObjC中，与alloc语义相反的方法是dealloc还是release？与retain语义相反的方法是dealloc还是release，为什么？需要与alloc配对使用的方法是dealloc还是release，为什么？   
22、在一个对象的方法里面:  
self.name = “object”;和  
name ＝”object”  
有什么不同吗?   
23、这段代码有什么问题吗:  
@implementation Person  
- (void)setAge:(int)newAge {  
self.age = newAge;  
}  
@end  
24、以下每行代码执行后，person对象的retain count分别是多少  
Person \*person = [[Person alloc] init];  
[person retain];  
[person release];  
[person release];   
25、autorelease的对象是在什么时候被release的？   
26、这段代码有什么问题,如何修改  
for (int i = 0; i < someLargeNumber; i++) {   
NSString \*string = @”Abc”;  
string = [string lowercaseString];  
string = [string stringByAppendingString:@"xyz"];  
NSLog(@“%@”, string);  
}   
27、autorelease和垃圾回收机制(gc)有什么关系？   
28、什么是Notification？   
29、什么时候用delegate，什么时候用Notification？   
30、什么是KVC和KVO？   
31、Notification和KVO有什么不同？   
32、KVO在ObjC中是怎么实现的？   
33、线程理解，有什么好处  
34、考察对@interface与@propety的理解  
35、objective-c中的类型转换分为哪几类  
36、多态的理解  
  
刘文(78769310) 2013/9/10 0:05:26  
这么多