

内存管理问题总结 共54个问题

以问题驱动学习的好处

1. 学习效率极其的高
2. 能够清楚的知道,自己那里会了,那里不会。(时间)
3. 课后写代码练习时目的性很强
4. 擅长总结,在面试的时候能够体现的淋漓尽致,使你面试非常容易
5. 方便日后复习
6. 只要你能够把每天的问题都能够通过自己的总结清晰的搞明白,那么你的薪水一定是10k以上

**内存管理 问题

1. 什么是ios内存管理?

就是在对象不再被使用的时候,把它即时的从内存中清除掉

2. 为什么要使用内存管理?

1. 严格的内存管理,能够是我们的应用程在性能上有很大的提高
2. 如果忽略内存管理,可能导致应用占用内存过高,导致程序崩溃

3. 系统判断一个对象是否要被销毁的依据是什么?

每个对象创建出来的时候,都有一个retainCount属性,默认值是1,当retainCount = 0的时候,系统就会将该对象销毁

4. 如何使对象的retainCount 值增加?

调用retain 对象方法

5. 如何使对象的retainCount 值减少?

调用release 对象方法

6. 如何判断对象已经被销毁了?

1. 从写NSObject提供的,dealloc方法,当对象即将被销毁的时候,默认会调用该方法
2. dealloc方法中一定要调用[super dealloc]方法
- 6.1 dealloc这个函数是那里来呀?

7. 内存管理原则是什么?

只要是出现new, alloc, retain,就要配对出现release操作,或者autorelease操作

**单个对象内存管理 问题

1. 什么是野指针?

对象的retainCount已经为0,保存了对象指针地址的变量就是野指针

1.1 使用野指针会有什么问题?

使用野指针调用对象的方法,会导致野指针异常,导致程序直接崩溃

2. 什么是僵尸对象?

retainCount = 0的对象被称之为僵尸对象,也就是不能够在访问的对象

2.1 是什么问题导致,访问僵尸对象,时而正确时而错误?

2.2 如何开始xcode的时时检测僵尸对象功能?

3. 如何防止出现野指针操作?

通常在调用完release 方法后,会把保存了对象指针地址的变量清空,赋值为nil
在oc中没有空指针异常,所以使用[nil retain]调用方法不会导致异常的发生

4. 什么是内存泄漏?

已经不在使用的对象,没有正确的释放掉,一直驻留在内存中,我们就说是内存泄漏

5. 内存泄漏有几种情况?

1. 没有配对释放,不符合内存管理原则
2. 对象提前赋值为nil或者清空,导致release方法没有起作用

6. 当对象的retainCount = 0 时 能否调用 retain方法使对象复活?

已经被释放的对象是无法在复活的

7. 关于内存我们主要研究的问题是什么?

1. 野指针
2. 内存泄露

**多个对象内存管理 问题

1. 对象与对象之间存在几种关系?

1. 继承关系
2. 组合关系
3. 对象作为方法参数传递

2. 对象的组合关系中,如何确保作为成员变量的对象,不会被提前释放?

重写set方法,在set方法中,retain该对象,使其retainCount值增加 1

3. 组合关系导致内存泄漏的原因是什么?

在set方法中,retain了该对象,但是并没有配对释放

4. 作为成员变量的对象,应该在那里配对释放?

在dealloc函数中释放

** set方法内存管理 问题

1. 在对象的组合关系中,导致内存泄漏有几种情况?

1. set方法中没有retain对象
2. 没有release掉旧的对象
3. 没有判断向set方法中传入的是否是同一个对象

2. 该如何正确的重写set方法?

1. 先判断是否是同一个对象
2. release一次旧的对象
3. retain新的对象

**内存管理@property参数 问题

1. @property参数分为几类?

1. 与set方法内存管理相关的参数

2. 是否要生成set方法相关
3. 多线程相关
4. set和get方法的名称相关

2. @property参数那些适用于对象类型,那些适用于基本数据类型

3. 如何使用@property生成符合内存管理的set方法?

4. @property retain参数能否用于基本数据类型?

****手动内存管理类的循环引用 问题**

1. 什么情况下会出现类的循环应用?
2. @class关键字的作用?
3. 手动内存管理如何解决类的循环引用问题?
4. 在.h文件中使用@class关键字声明一个类后,在.m文件中是否能够直接调用该对象的方法?

****自动释放池 问题**

1. 什么是自动释放池?
自动释放池是用来存储多个对象类型的指针变量
2. 自动释放池对池内对象的作用?
被存入到自动释放池内的对象,当自动释放池被销毁时,会对池内的对象全部做一次release操作
3. 对象如何放入到自动释放池中?
当你确定要将对象放入到池中的时候,只需要调用对象的 autorelease 对象方法就可以把对象放入到自动释放池中
4. 如何创建自动释放池?

对象在自动释放池内部调用autorelease 方法

5. 自动释放池能否嵌套使用?
能

6. 自动释放池何时被销毁?
在autorelease } 执行完后

7. 多次调用对象的autorelease方法会导致什么问题?
多次将地址存到自动释放池中,导致野指针异常

8. 自动释放池作用
将对象与自动释放池建立关系,池子内调用autorelease 方法,在自动释放池销毁时销毁对象,延迟release销毁时间

****自动释放池应用 问题**

1. 实际开发中一般如何使用autorelease

就是在方法中创建新的对象并且需要返回的时候
快速创建一个类方法

**ARC机制问题

1. 什么是ARC机制

自动引用计数, 不需要程序员关心, 对象的retain, 与release操作

2. 什么是强指针(strong), 弱指针(weak)

3. ARC机制中, 系统判断对象是否被销毁的依据是什么?

指向对象的强指针是否被销毁

4. ARC机制中, 如何观察对象被销毁了?

5. ARC机制中为什么还有autoreleasepool?

6. ARC机制的本质是什么?

对releaseCount的计算, 创建对象 +1, 清空指针 -1, 或者越到autoreleasepool的大括号 -1

7. ARC的目的

是让程序员不在关心对象的retainCount

**ARC机制 @property参数 问题

1. ARC机制中如何让@property生成符合内存管理的set方法

2. @property weak参数能否用于基本数据类型?

不能

3. 为什么在ARC机制中, 不建议使用assign类表示对象的直接赋值

4. ARC机制中不建议使用retain,assign, 容易造成混淆

**手动内存管理代码转换成ARC代码 问题

1. 如何使用xcode自带的功能, 将手动内存管理代码转换成ARC机制代码

手动内存管理与ARC机制代码共存 问题

1. 为什么会出现手动内存管理与ARC机制代码共存现象?

2. 为什么不统一的将代码都转换成ARC机制?

3. 如何才能够让手动内存管理的代码与ARC机制的代码共存?

**ARC机制中类的相互引用 问题

1. ARC机制中类的相互引用,与手动内存管理类的相互引用有什么区别吗?

2. 如何解决ARC机制下类的相互引用问题?

在.h文件中使用@class关键字声明一个类,两端不能都用强指针
一端用strong,一端用weak