

1. Object-C 有多继承吗？没有的话用什么代替？

- 1> OC 是单继承，没有多继承
- 2> 有时可以用分类和协议来代替多继承

2. Object-C 有私有方法吗？私有变量呢？

- 1> 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
- 2> 可以使用类扩展（Extension）来增加私有方法和私有变量

3. 关键字 const 什么含义？

```
const int a;  
int const a;  
const int *a;  
int const *a;  
int * const a;  
int const * const a;
```

- 1>前两个的作用是一样：a 是一个常整型数
- 2>第三、四个意味着 a 是一个指向常整型数的指针(整型数是不可修改的，但指针可以)
- 3>第五个的意思：a 是一个指向整型数的常指针(指针指向的整型数是可以修改的，但指针是不可修改的)
- 4>最后一个意味着：a 是一个指向常整型数的常指针(指针指向的整型数是不可修改的，同时指针也是不可修改的)

4. static 的作用？

- 1> static 修饰的函数是一个内部函数，只能在本文件中调用，其他文件不能调用
- 2> static 修饰的全部变量是一个内部变量，只能在本文件中使用，其他文件不能使用
- 3> static 修饰的局部变量只会初始化一次，并且在程序退出时才会回收内存

5. 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

6. 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

7. 为什么很多内置的类,如 **UITableView** 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

8. 定义属性时，什么情况使用 **copy**、**assign**、**retain**?

- 1> copy: NSString、Block 等类型
- 2> assign: 非 OC 对象类型，基本数据类型（两个对象相互引用的时候，一端用 retain，一端用 assign）
- 3> retain: OC 对象类型

9. 对象是什么时候被释放的?

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

10. 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）

11. ViewController 的 **loadView**、**viewDidLoad**、**viewDidUnload** 分别是什么时候调用的，在自定义 **ViewCointroller** 时在这几个函数中应该做什么工作?

- 1> loadView
 - 当第一次使用控制器的 view 时，会调用 loadView 方法创建 view
 - 一般在这里自定义 view
- 2> viewDidLoad
 - 当控制器的 view 创建完毕时会调用，也就是在 loadView 后调用
 - 一般在这里添加子控件、初始化数据

3> viewDidUnload

- 当控制器的 view 因为内存警告被销毁时调用
- 一般在这里回收跟界面相关的资源（界面都会销毁了，跟界面相关的资源肯定不要了）

12. ViewController 的 didReceiveMemoryWarning 是在什么时候调用的？默认的操作是什么？

- 当应用程序接收到系统的内容警告时，就有可能调用控制器的 didReceiveMemoryWarning 方法
- 它的默认做法是：
 - 当控制器的 view 不在窗口上显示时，就会直接销毁，并且调用 viewDidUnload 方法

13. 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）

14. self跟 self->什么区别？

- 1> self是调用 get 方法或者 set 放
- 2> self 是当前本身，是一个指向当前对象的指针
- 3> self->是直接访问成员变量

15. id、nil 代表什么？

- 1> id 类型的指针可以指向任何 OC 对象
- 2> nil 代表空值（空指针的值， 0）

16. 如何对 iOS 设备进行性能测试？

Timer Profile