1. 简述OC中内存管理机制。与retain配对使用的方法是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？readwrite，readonly，assign，retain，copy，nonatomic 、atomic、strong、weak属性的作用？   
   OC使用了一种叫做引用计数的机制来管理对象，如果对一个对象使用了alloc、[Mutable]copy，retain，那么你必须使用相应的realease或者autorelease。也可以理解为自己生成的对象，自己持有。非自己生成的对象，自己也能持有。不在需要自己持有的对象时释放。非自己持有的对象无法释放。生成并持有对象<alloc,new,copy,mutableCopy等>，持有对象<retain>，释放对象<release>,废弃对象<dealloc>。readwrite(默认):可读可写，表示既有getter方法，也有setter方法。readonly:表示只有getter方法，没有setter方法。nonatomic:不考虑线程安全。atomic（默认）：线程操作安全。strong（默认）:ARC下和MRC下retain一样，weak（ARC下）：和(MRC下)assign类似，区别是当weak指向的内存释放掉后自动置为nil，防止野指针。

unsafe\_unretained声明一个若引用，但不会自动置为nil，可能会出现野指针。

线程安全下的setter和getter方法：

* （NSString \*）value{

@synchronized（self）{

return [[\_value retain] autorelease];

}

}

* (void)setValue:(NSString \*)aValue{

@synchronized(self){

[aValue retain];

[\_value release];

\_value = aValue;

}

}

1. 类变量的@protected ,@private,@public,@package，声明各有什么含义？  
   上面的几个声明表明的时类成员的作用域，@private作用范围只能在自身类（外界既不可访问，又不能继承）；@protected作用范围在自身类和子类，如果什么都不加修饰，默认是@protected（外界不可访问，但是可以继承）；@public作用范围最大，可以在任何地方被访问（外界即可访问，又可以继承）；@package作用范围在某个框架内
2. 线程是什么？进程是什么？二者有什么区别和联系？   
   线程是CPU独立运行和独立调度的基本单位（可以理解为一个进程中执行的代码片段），进程是资源分配的基本单位（进程是一块包含了某些资源的内存区域）。进程是线程的容器，真正完成代码执行的是线程，而进程则作为线程的执行环境。一个程序至少包含一个进程，一个进程至少包含一个线程，一个进程中的多个线程共享当前进程所拥有的资源。
3. 谈谈你对多线程开发的理解？ios中有几种实现多线程的方法？  
   好处：  
   1、使用线程可以把程序中占据时间长的任务放到后台去处理，如图片、视频的下载  
   2、发挥多核处理器的优势，并发执行让系统运行的更快、更流畅，用户体验更好  
   缺点：  
   1、大量的线程降低代码的可读性，  
   2、更多的线程需要更多的内存空间  
   3、当多个线程对同一个资源出现争夺的时候要注意线程安全的问题。  
   iOS有三种多线程编程的技术：  
   1、NSThread(两种创建方式)  
   [NSThread detachNewThreadSelector:@selector(doSomething:) toTarget:self withObject:nil];  
   NSThread \*myThread = [[NSThread alloc] initWithTarget:self selector:@selector(doSomething:) object:nil];

[myThread start];  
2、NSOperationQueue

NSOperationQueue \*oprationQueue = [[NSOperationQueue alloc] init];

oprationQueue addOperationWithBlock:^{

//这个block语句块在子线程中执行

}

http://alloc.sinaapp.com/wp/?p=237

3、[Grand Central Dispatch (GCD)](http://developer.apple.com/library/ios/#documentation/Performance/Reference/GCD_libdispatch_Ref/Reference/reference.html)

dispatch\_async(dispatch\_get\_global\_queue(DISPATCH\_QUEUE\_PRIORITY\_DEFAULT, 0), ^{

// 耗时的操作

dispatch\_async(dispatch\_get\_main\_queue(), ^{

// 更新界面

});

});

http://blog.csdn.net/totogo2010/article/details/8016129

PS:不显示的创建线程的方法：  
用NSObject的类方法  performSelectorInBackground:withObject: 创建一个线程： [Obj performSelectorInBackground:@selector(doSomething) withObject:nil];

1. 线程同步和异步的区别？IOS中如何实现多线程的同步？  
   同步：一个线程要等待上一个线程执行完之后才能执行当前的线程，生活中的例子（上厕所）。  
   异步：同时去做两件或者多件事。比如边听歌边看报。   
   GCD串行队列,GCD当中的屏障dispatch\_barrier,NSOperationQueue设置最大并发数为1

参考（<http://blog.csdn.net/lifengzhong/article/details/7487505>）

多线程实现数据：原子操作(atomic)、加锁（NSLock、NSRecursive、NSConditionLock）、@synchronized

1. 假设有一个字符串aabcad，请写一段程序，去掉字符串中不相邻的重复字符串，即上述字符串处理之后的输出结果为：aabcd

本题的题意是以第一个出现的字母作为参照，只要之后出现相同的字母并且和第一个字母不相邻，那么就删除。为防止删除某些字符之后，把之前不相邻的重复字符串转化为相邻字符串，所以可以先用空格替换掉需要删除的字符，最后对数组里面的空格进行处理。  
-(void)removeRepeat:(NSString \*)aNum

{

NSMutableArray \*mArr = [[NSMutableArray alloc]initWithCapacity:10];

for (int i = 0;i < aNum.length;i++)

{

[mArr addObject:[aNum substringWithRange:NSMakeRange(i,1)]];

}

NSLog(@"- %@", mArr);

[self compareNum:mArr];

NSLog(@"%@",mArr);

}

//比较是否相等

-(NSMutableArray \*)compareNum:(NSMutableArray \*)mArr

{

int count = mArr.count;//重新定义了,count不会减一

for (int j = 0; j < count - 1; j++)

{

for (int i = j; i < count - 1-1-1; i++)

{

NSLog(@" %@ %@",[mArr objectAtIndex:j],[mArr objectAtIndex:i + 2]);

NSString \*a = [mArr objectAtIndex:j];

NSString \*b = [mArr objectAtIndex:i+2];

if ([a isEqualToString:b])

{

[mArr replaceObjectAtIndex:i + 2 withObject:@" "];

}

}

}

return mArr;

}

1. 获取一台设备唯一标识的方法有哪些？  
   MAC地址，udid，keychain，open udid,广告标识IDFA-identifierForIdentifier

参考链接< http://www.2cto.com/kf/201308/237648.html >。

1. iOS类是否可以多继承？如果没有，那可以用其他方法实现吗？简述实现过程。  
   不可以，可以通过delegate和protocol和类别来实现类似多继承。
2. 堆和栈的区别？  
   栈区（stack）--由编译器自动分配释放，存放函数的参数值、局部变量的值。  
   堆区（heap）--一般由程序员分配释放。  
   全局区（静态区）（static）--全局变量和静态变量。程序结束后由系统释放。

文字常量区--常量字符串存放在这里。程序结束后由系统释放。  
 程序代码区—存放函数体的二进制文件。

栈：只要栈的剩余空间大于所申请空间，系统将为程序提供内存，否则将报异常提示栈溢 出。

堆：首先应该知道操作系统有一个记录空闲内存地址的链表，当系统收到程序的申请时，会遍历该链表，寻找第一个空间大于所申请空间的堆结点，然后将该结点从空闲结点链表中删除，并将该结点的空间分配给程序，另外，对于大多数系统，会在这块内存空间中的首地址处记录本次分配的大小，这样，代码中的delete语句才能正确的释放本内存空间。另外，由于找到的堆结点的大小不一定正好等于申请的大小，系统会自动的将多余的那部分重新放入空闲链表中。

1. iOS本地数据存储都有哪几种方式？iOS如何实现复杂对象的存储？
2. NSKeyedArchiver（归档）采用归档的形式来保存数据，该数据对象需要遵守NSCoding协议，并且该对象对应的类必须提供encodeWithCoder:和initWithCoder:方法。
3. NSUserDefaults：用来保存应用程序设置和属性、用户保存的数据。用户再次打开程序或开机后这些数据仍然存在。NSUserDefaults可以存储的数据类型包括：NSData、NSString、NSNumber、NSDate、NSArray、NSDictionary。
4. Write写入方式：永久保存在磁盘中。
5. SQLite（FMDB、CoreData）

NSCoding + NSKeyedArchiver实现复杂对象的存储。

1. iOS的动态性  
   iOS的动态性来自三个方面：动态类型、动态绑定、动态载入、SEL类型

1、动态类型<弱类型>（id）：在代码的运行阶段判断代码的类型，使用id类型可以让应用在“运行时”使用任何类型来替换。动态类型让程序更加灵活，但是会使数据的统一性降低和代码的可读性。我们常用静态类型<强类型>（如NSString）,使用静态类型编译器可以完全分析你的代码，这让代码的性能和可预知性更高。  
2、动态绑定：让代码在运行时判断需要调用什么方法，而不是在编译时。 动态类型和动态绑定使得选择哪个接收者已经调用什么方法都放到运行  
 时去完成。 3、动态载入：应用程序可以根据需要加载可执行代码以及资源，而不是

在启动时就加载所有资源。

4、SEL类型 iOS在编译的时候会根据方法的名字（包括参数序列），生成一个用来区分这个方法的唯一的ID，这个ID是SEL类型的，SEL的本质就是类方法的编号[函数地址]。（类似C语言里面的函数指针，但是OC的类不能直接使用函数指针，这样只能做一个@selector语法来取。注意：@selector是查找当前类（含子类）的方法。）