

IOS 相关基础知识整理

1. Object-C 的类不可以多继承，可以实现多个接口。重写一个类的方式用分类比用继承好，因为分类可在不熟悉，不改变原代码的情况下往类中添加方法，而继承需对父类很熟悉，耦合性太强。
2. #import 是 Object-C 导入头文件的关键字，#include 是 C/C++ 导入头文件的关键字，使用#import 头文件会自动只导入一次，不会出现重复引用：例如：ClassA 与 ClassB 同时引用了 ClassC，不做重复引用处理的时候在 ClassD 中同时引用 ClassA，ClassB 编译会提示对 ClassC 重复引用的错误。@class 向前声明，告诉编译器某个类的声明，当执行时，才去查看类的实现文件，可以解决头文件相互包含问题，在.m 文件中#import。#import<>导入框架，#import"" 导入某个头文件。
3. 属性 readwrite 是可读写特性，需要生成 getter 和 setter 方法时使用；readonly 是只读属性，只会生成 getter 方法，不会生成 setter 方法，在不希望属性在类外被改变时使用；assign 是赋值特性，仅设置变量时使用，也可修饰对象，但若对象释放了，指针指向对象的内存还在，成为野指针，如果之后再往该对象发送消息会崩溃；retain 表示持有特性，setter 方法将传入参数先保留，再赋值，传入参数的 retaincount 会+1；copy 表示赋值特性，setter 方法将传入对象复制一份，需要完全新的一份新的变量时使用；nonatomic 表示非原子操作，决定编译器的 setter getter 是否是原子操作，atomic 表示多线程安全，一般使用 nonatomic

4. - (void)setName:(NSString *)str{

```
[str retain];
```

```
[name release];
```

```
name = str;
```

```
}
```

```
-(void)setName:(NSString *)str{
```

```
id t = [str copy];
```

```
[name release];
```

```
name = t;
```

```
}
```

5. 对于语句 `NSString *obj = [[NSData alloc] init];` `obj` 在编译时是 `NSString` 类型；运行时是 `NSData` 类型的对象
6. Object-C 的数据类型有 `NSString`, `NSNumber`, `NSArray`, `NSData` 等等, 这些都是 `class`, 创建后便是对象, 而 `c` 语言等基本数据类型 `int` 知识一定字节的内存空间, 用于存放数值; `NSInteger` 是基本数据类型 `int` 或者 `long` 的别名, 它的区别在于会根据系统是 32 位还是 64 位来决定本身是 `int` 还是 `long`。
7. `id` 声明的对象具有运行时的特性, 可以指向任何类型的 `objective-c` 的对象。
8. Object-C 的内存管理主要有三种方式 `ARC` (自动引用计数)、`MRC` (手动引用计数)、`autoreleasepool` (自动释放池);
9. `atomic` 提供多线程安全。是防止在写未完成的时候被另一个线程读区, 造成数据错误
`nonatomic` 只是简单地返回这个值而已
10. 内存管理的原则: 谁申请谁释放; 谁引用谁管理
关键字 `alloc` 或 `new` 或 `copy` 出来的对象需要手动释放, 设置正确的 `property` 属性, 对于 `retain`、`strong` 对象需要在何时的地方释放
11. 通过 `Profile`→`instruments`→`time profiler` 可以对 `ios` 设备进行性能测试。
12. `object-c` 中创建线程的方法
`[object performSelectorInBackground:];`
`[NSThread detachNewThreadSelector:];`
`[[NSThread alloc] initWithTarget:] start];`
`[[NSInvocationOperation alloc] initWithTarget...] start];`
`[NSBlockOperation blockOperationWithBlock:^()];`

```
[_queue addOperation:];
```

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT  
, 0, ^ {}));
```

在主线程中执行代码的方法

```
[object performSelectorOnMainThread:];
```

```
[NSOperationQueue mainQueue] addOperationWithBlock:^ {};
```

```
dispatch_async(dispatch_get_main_queue(), ^ {});
```

延时执行代码

```
[object performSelector:withObject:afterDelay:];
```

```
dispatch_after(DISPATCH_TIME_NOW, (int64_t) delayInSeconds *  
NSEC_PER_SEC), dispatch_main_queue(), ^ {});
```

13. MVC 设计模式：模型、视图控制器，可以将整个应用程序在思想上分为三大块，对应是数据的存储或处理，前台的显示，业务逻辑的控制。iPhone 本身的设计思想就是遵循 mvc 设计模式，其不属于 23 中设计模式范畴。

代理模式：给某个对象提供一个代理对象，这个对象可以指示代理对象执行任务。

单例模式：在程序运行期内只被实例化一次的类叫单例，属于系统级全局对象，有利于协调系统整体的行为。

观察者模式：当一个物体发生变化时，会通知所有观察这个物体的观察者让其做出反应。实现起来无非就是把观察者的对象给这个物体，当物体发生改变时遍历观察者的对象，调用观察者的监听方法从而达到通知观察者的目的。

工厂模式：隐藏实现细节，才用静态方法，如 NSNumber 的实例化操作。

14. 浅复制只复制对象的指针，而不复制引用对象本身；深复制复制引用对象本身；浅复制好比你和你的影子，你消失，影子也消失；深复制好比你和你的克隆人，你消失，它还在。
15. category 可以不获悉，不改变原来代码的情况下往里面添加新的方法，只能添加，不能删除修改，并且如果类别和原来类中的方法产生命名冲突，则类别将覆盖原来的方法，因为类别具有更高的优先级。类别主要有三个作用：（1）将类的实现分散到多个不同文件或不同框架中去（2）创建私有方法的前向引用（3）向对象添加非正式协议。继承可以增加、修改或删除方法，并且可以增加属性。

16. category 和 extensions 的不同在于后者可以添加属性。另外，后者添加的方法是必须要实现的。Extensions 可以认为是一个私有的 category

17. oc 中的代理有 2 层含义, formal 和 informal protocol。前者和 java 接口一样。Informal protocol 其实就是类别的另一种表达方式” 这里有一些你可能希望实现的方法，你可以使用他们更好的完成工作”。非正式协议使用 interface 修饰，@required 必须实现，@optional 可选实现。

18. kvc: 键-值编码是一种间接访问对象的属性，使用字符串来标识属性，而不是通过调用存取方法，直接访问实例变量的机制。很多情况下可以简化程序代码。

Kvo: 键值观察机制，提供了观察某个属性变化的方法，极大简化了代码。比如监听按钮点击变化

```
[self addObserver:self forKeyPath:@"highlighted" options:0 context:nil];
```

```
- (void)observeValueForKeyPath:ofObject:change:context:{
```

```
    if([keyPath isEqualToString:@"highlighted"]){
```

```
        //略
```

```
    }
```

```
}
```

根据 keypath 去取到相应值的变化，理论上是和 kvc 道理一样的

根据 key 寻找 value 的过程：（1）查找对象是否有 valueForKey 这个方法（2）查找对象是否带有 valueForKey 这个实例变量，如果没找到，会继续调用-valueForKey:这个方法，如果这个方法还是没有被实现的话，程序会抛出一个 NSUndefinedKeyException 异常错误。

19. 代理的目的是改变或传递控制链。允许一个类在特定时刻通知到其他类，而不需要获取到那些类的指针，可以减少框架复杂度。aa

20. NSArray 和 NSMutableArray。前者在初始化后的内存空间就是固定不变的，而后者可以添加操作等，动态申请新的内存空间。

21. 说 oc 是动态运行时语言，主要是将数据类型的确认由编译时推迟到了运行时。运行时机制使我们直到运行时才去决定一个对象的类别，以及调用哪个方法。
22. 不同对象以自己的方式响应相同消息的能力叫做多态
23. 协议由控制链 has-a 关系，体现单一拥有和可控制的对应关系，而通知没有，代理一对一，一条消息发给一个消息接收者，而通知可以一对多，一条消息发给多个消息接收者。
24. 推送消息是解决轮询所带来的流量浪费和电量消耗问题，直接由服务端主动 push 消息到客户端。
25. objective-c 中要实现一个单例类，至少需要做以下四个步骤：
 - (1) 为单例对象实现一个单例类，并初始化，然后设置为 nil；
 - (2) 实现一个实例构造方法检查上面声明的静态实例是否为 nil，如果是则创建并返回一个本类的实例
 - (3) 重写 allocWithZone 方法，来保证其他人直接使用 alloc 和 init 视图获得一个新实例的时候不产生一个新实例
 - (4) 适当实现 allocWithZone, copyWithZone 等方法
26. 响应链式通过递归构成的一组 UIResponder 对象的链式序列，
application<-viewController<-view1<-view2<-button，按钮点击后，递归查询能够响应点击事件的对象，如果 application 不能响应，那么事件会被丢弃。
27. frame 标示该 view 在父视图坐标系中的位置和大小，bounds 标示该 view 在自身坐标系中的位置和大小
28. selector 是一个方法的名字，而 method 是一个组合体，包含了名字和实现
29. oc2.0 有 Garbage collection，但是 iOS 平台不提供

30. `NSOperationQueue` 是存放 `NSOperation` 的集合类。虽然是 queue，但是却不带有队列的概念，放入的操作并非是严格按照先进先出。`MYNetwork` 网络库中的 `MYTaskOperationQueue` 也是以此为参照的，总之，它可以帮助进行多线程编程。
31. 延时加载又称懒加载，最好的例子就是 `tableView` 中的图片加载，一个延时加载避免内存过高，一个异步加载，避免线程阻塞。
32. 一般来说，一个视图控制器提供一个视图，视图不能放控制器如 `tableView` 控制器，但可以通过 `[controller addChildController:]` 实现。`MYSliderView` 就是以此为依据进行封装的。
33. 一个 `tableView` 可以关联两个不同的数据源，不过得把握时间进行相应数据源的切换。
34. 当数组在程序运行时，需要不断变化的，使用 `NSMutableArray`，当数组在初始化后，便不再改变的使用 `NSArray`。`NSArray` 是线程安全的，`NSMutableArray` 不是线程安全的，多线程使用到 `NSMutableArray` 需要多加注意。
35. `UITableView` 的 Data Source 方法-`tableView:numberOfRowsInSection:`
-`tableView:cellForRowAtIndexPath:`
36. 界面线程维护着自己的内存池，用户自己创建的数据线程，需要创建该线程的内存池。
37. 简便构造器一般由 `cocoaTouch` 框架提供，如 `NSNumber` 的 `numberWithBool` 等等。即工厂方法创建对象。获得的对象不需要手动释放。
38. `iPhone` 应用保存数据的方式有：(1)通过 web 服务，保存在服务器上(2)通过 `NSCoder` 固化机制，将对象保存在文件中(归档)(3)通过 `SQLite` 或 `CoreData` 保存在文件数据库中

39. CoreData 是苹果提供一套数据保存框架，基于 SQLite
40. NSManagedObject 是 NSObject 的子类，也是 CoreData 的重要组成部分，是一个通用的类，实现了 CoreData 模型层所需的基本功能，用户可以通过子类化 NSManagedObject，建立自己的数据模型。
41. NSManagedObjectContext 对象负责应用和数据库之间的交互
42. 谓词是 NSPredicate，通过给定的逻辑条件完成对数据的筛选。如：

```
predicate = [NSPredicate predicateWithFormat:" customerID==%d",n];  
a = [customers filteredArrayUsingPredicate:predicate];
```
43. 持久化存储方式包括：存到文件 plist、NSUserDefaults、归档、存到 sqlite 文件数据库、存到 CoreData
44. block 是可以获取其他函数内部变量的匿名函数，不但方便开发，而且可以大幅度提高应用的执行效率(多核 cpu 可直接处理 block 指令)

```
typedef void (^animations)(void);  
typedef void(^completion)(BOOL finished);
```
45. NSURLConnection 主要用于网络访问，其中
+sendSynchronousRequest:returningResponse:error: 是同步访问数据，即当前线程会阻塞，并等待 request 返回的 response，而 initWithRequest:delegate:使用的是异步加载，当其完成网络访问后，会通过 delegate 回到主线程，并通知其委托的对象。
46. 多线程按字面的意思是同步完成多项任务，提高系统资源的使用效率。多线程可以让应用有更快的回应，可以在执行耗时操作的时候，同时相应用户的触摸操作。在 ios 应用中，对多线程最初的理解，就是并发。

47. GCD 是从系统级别提供的一个易用的多线程类库，具有运行时的特点，能充分利用多核处理器的优势。接口为 C 语言。函数参数中有 Block，为我们提供强大的接口
- NSOperation 是一个抽象类，封装了线程的细节，可以通过子类化对象，加上 NSOperationQueue 面向对象的思维，管理多线程程序。
- NSThread 是一个控制线程执行的对象，它不如 NSOperation 抽象，通过它我们可以方便的得到一个线程，并控制它。但 NSThread 的线程之间的并发控制需要自己来，可以通过 NSCondition 实现。
- 通知、Timer 和异步任务等都有使用到多线程
48. 项目中使用 NSOperation 的优点是 NSOperation 是对线程的高度抽象，在项目中使用它，会使项目的程序结构更好，子类化 NSOperation 的设计思路，具有面向对象的优点(复用、封装)，使得实现是多线程支持，而接口简单，建议在复杂项目中使用。
- 项目中使用 GCD 的优点是本身非常简单、易用，对于不复杂的多线程操作会节省代码量，而 block 参数的使用使代码更为易读，建议在简单项目中使用 /
49. 声明一个 block，注意对象属性设置为 copy，接到 block 参数时，便会自动复制一份到堆上。__block 是一种特殊类型，用该关键字声明的局部变量，可以被 block 所改变，并且其在原函数中的值会被改变。使用 block 实现委托模式的优点是，其回调定义在委托对象函数内部，使代码很简凑易读，代码更简洁。
50. 当你使用 new, alloc 和 copy 方法创建一个对象时，该对象的保留计数器值为 1。当你不再使用该对象时，你要负责向对象发送一条 release 消息或 autorelease 消息，这样，对象将在使用结束时被销毁；当通过任何其他方法获取一个对象时，不许手动 release，但如果想一段时间内持有该对象，需要保留它，而且在操作完成时释放它。
51. 在 Objective-C 中，所有实例变量默认都是私有的，所有实例方法默认都是公有的。可在分类中实现私有方法
52. Object-C 没有多继承，使用委托代理来代替，多态特性在 obj-c 中主要通过委托来实现的。

53. NSArray 对象会 retain+1 任何数组中的对象。当 NSArray 被 dealloc 时候，所有数组中的对象会执行一次 retain-1。不仅仅是 NSArray，人和收集类都执行类似操作，例如 NSDictionary，甚至 UINavigationController。

54. Obj-c 的编译器处理后缀为 m 的文件时，可以识别 obj-c 和 c 的代码，处理 mm 文件时可以识别 obj-c，c，c++的代码。但 cpp 文件必须只能用 c/c++代码，且 cpp include 的头文件不能出现 obj-c 的东西。如何在 cpp 混用 objc？

55. 管理方式：对于栈来讲，是由编译器自动管理，无需我们手工控制；对于堆来说，释放工作由程序员来控制，容易产生 memory leak。

栈：是向低地址扩展的数据结构，是连续的一块内存的区域，栈顶的地址和栈的最大容量是系统预先规定好的，如果申请的空间超过栈道容量，将提示 overflow。因此，能从栈获得的空间较小。

堆：堆是向高地址扩展的数据结构，是不连续的内存区域。这是由于系统是用链表来存储空闲地址的，自然是不连续的，而链表的遍历方向是由低地址向高地址。堆堆获得的空间比较灵活，比较大。

对于堆来讲，频繁的 new / delete 势必造成空间的不连续，从而造成大量的碎片，使程序运行效率降低。而对于栈来讲，则不会存在这个问题，因为栈是先进后出的队列，不可能有一个内存块从栈中间弹出。

56. delegate 真对一对一关系，用于 sender 接收到 receiver 的某个反馈值
notification 真对一对多关系，用于通知多个 object 某个事件。

57. 宏定义：#define SECONDS_PER_YEAR (60*60*24*365)UL，表示一年中有多少秒
宏定义：#define MIN(A,B) ((A)<(B)?(A):(B))

58. 关键字 const 意味着只读，const int a;或 int const a;表示一个常整型数。
Const int *a;表示一个指向常整型数的指针。Int *const a;是一个指向整型数的常指针，a

可以变指针不能变，如 static NSString *const a;

59. 函数体内 static 变量的作用范围为该函数体，不同于 auto 变量，该变量的内存只被分配一次，下次调用还是维持上一次的值

在模块内的 static 全局变量可以被模块内所有函数访问，但不能被模块外其他函数访问。

在类中的 static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝，这个函数不接受 this 指针，因而只能访问类的 static 成员变量。

60. http 是客户端用 http 协议进行请求，发送请求时需要封装 http 请求头，并绑定请求的数据，服务器一般由 web 服务器配合。http 请求方式为客户端主动发起请求，服务器才能给响应，一次请求完毕后断开连接，以节省资源。服务器不能主动给客户端响应，除非采用 http 长连接技术。

Socket 是客户端和服务端直接使用 socket 套接字进行连接，没有规定连接后断开，所以客户端和服务端可以保持连接通道，双方都可以主动发送数据，一般在游戏开发或股票开发这种要求即时性比较强并且保持发送数据量比较大的场合使用。主要使用类是 CFSocketRef。

61. TCP 全称是 Transmission Control Protocol，中文名为传输控制协议，它可以提供可靠的、面向连接的网络数据传递服务。传输控制协议主要包含以下任务：确保 IP 数据报的成功传递；对程序发送的大块数据进行分段和重组；确保正确排序及顺序传递分段的数据；通过计算校验和，进行传输数据的完整性检查。TCP 提供的是面向连接的、可靠的数据流传输，而 UDP 提供的是非面向连接的、不可靠的数据流传输。简单的说 TCP 注重数据安全，而 UDP 数据传输快点，但安全性一般

62. LoadView 在 controller 的 view 为 nil 时候调用，此方法中可设置自定义 view。

63. 代码区：存放函数二进制代码

全局(静态区)：全局变量和静态变量，程序结束后由系统释放，全局区又细分为未初始化全局区和初始化全区域。

常量区：如常量字符串，程序结束后由系统释放

堆区：通过 malloc 等函数或 new 操作动态申请得到，需要程序员手动申请和释放

栈区：函数模块内申请，函数结束后由系统自动释放。存放局部变量，函数参数

64. 队列和栈是两种不同的数据容器，队列是种先进先出的数据结构，栈是种先进后出的数据结构。
65. HTTP 协议中，post 和 get 的区别：Get 方法提交数据不安全，数据置于请求行，客户端可见；Get 方法提交的数据大小有限；Get 方法不可设置标签。Post 方法提交的数据安全，数据置于消息主体内，客户端不可见；Post 方法提交的数据大小无限；Post 方法可以设置标签。
66. iOS 的系统架构从下到上为核心操作系统层、核心服务层、媒体层、界面服务层
67. 控件主要响应三种事件：基于触摸的事件、基于值的事件、基于编辑的事件
68. 动画分为隐式动画和显式动画
69. Core Animation 系统提供的过渡类型：交叉淡化、推挤、显示、覆盖
70. Quartz2d 的绘图功能的三个核心概念：上下文，主要用于描述图形写入哪里；路径，是在图层上绘制的内容；状态，保存配置变换的值、填充和轮廓，alpha 等。
71. iPhone OS 主要提供四种音频播放方法：SystemSound Service、AVAudioPlayer 类、Audio Queue Service、OpenAL
72. 使用 AVAudioPlayer 类调用的框架是 AVFoundation，步骤：配置 AVAudioPlayer 对象、实现 AVAudioPlayer 类的委托方法、控制 AVAudioPlayer 类的对象、监听音量水平；回放进度和拖拽播放。
73. 手势通知方法：
- touchesBegan:withEvent:
 - touchesMoved:withEvent:
 - touchesEnded:withEvent:

-touchesCanceled:withEvent:

74. CFSocket 使用步骤：创建 Socket 上下文、创建 Socket；配置要访问的服务器信息；封装服务器信息；连接服务器。
75. Core Foundation 中提供的操作 socket 的方法有：CFNetwork、CFSocket、BSD Socket
76. 解析 XML 文件方式：以 DOM 方式解析 XML 文件；以 SAX 方式解析 XML 文件
77. iOS 中可以有四种持久化存储的方式，属性列表 plist、对象归档、sqlite3 和 coredata；core data 可以使你以图形界面的方式快速定义 app 的数据模型，同时也在你的代码中容易获取到它。Core data 提供了基础结构去处理常用的功能，例如保存，恢复，撤销和重做，允许你在 app 中继续创建新的任务。在使用 core data 的时候，你不用安装额外的数据库系统，因为 core data 默认使用内置的 sqlite 数据库。Coredata 将你 app 的模型层放入到一组定义在内存中的数据对象。Coredata 会把这些对象归档，并永久性保存。Sqlite 库是一个轻量级功能强大的关系数据引擎，也很容易嵌入到应用程序，可以多个平台使用，sqlite 饰一个轻量级的嵌入式 sql 数据库编程。与 coredata 不同的是，sqlite 是使用程序式的 sql 的主要 api 来直接操作数据表。Coredata 不是一个关系型数据库，也不是关系型数据库管理系统。虽然 coredata 支持 sqlite 作为一种存储类型，但它不能使用 sqlite 任意数据库。Coredata 在使用的过程中自己创建这个数据库，支持一对一一对多的关系。