移动端开发规范

# Android开发规范

### 1.对项目的要求

1.1 请一定使用**Java**开发，不允许使用**Kotlin**

1.2 使用**MVP模式**开发

1.3 网络请求框架请使用 **okhttp / Retrofit**

1.4 如果有图片加载请使用 **Glide**

1.5 播放视频控件如果使用第三方开源库，请使用**ijkplayer**

1.6 需要适配**Pad**设备

1.7 项目需要做成库，最终交付项目可以使用gradle形式接入

android开发版本限制：

compileSdkVersion: 28,  
buildToolsVersion: "28.0.3",  
minSdkVersion : 19,  
targetSdkVersion : 29,  
supportVersion : "28.0.0"

### ****2.命名规范****

##### 1.包（packages）

* 采用反域名命名规则，全部使用小写字母。

例：一级包名为com，二级包名为google，三级包名为play，四级包名为模块名或层级名(model)，五级为业务细化(common)。 **com.itutorgroup.\***\*\*\*

##### 2.类（classes）

* 名词，采用大驼峰命名法，尽量避免缩写，除非该缩写是众所周知的， 比如HTML,URL，如果类名称中包含单词缩写，则单词缩写的每个字母均应大写。
* 若有全称太长的情况，优先使用约定俗成的缩写，其次使用每个单词的大写，这种情况请少出现。例：PstMainFragment ---> PagerSlidingTabMainFragment

例：CarFragment、BaseFragmentActivity

##### 3.接口（interface）

* 命名规则与类一样采用大驼峰命名法，多以able结尾。

例：Viewable、Populatable、Runnable 、Accessable 。

* 也可以使用大写的I开头，第二位英文首字母也大写

例：IBaseView、IPresenter

##### 4.方法（methods）

* 动词或动名词，采用小驼峰命名法例如:onCreate(),run()

例：

initXX() | 初始化相关方法,使用init为前缀标识，如初始化布局initView()

isXX() | checkXX()方法返回值为boolean型的请使用is或check为前缀标识

getXX() | 返回某个值的方法，使用get为前缀标识

##### 5.变量（variables）

* 采用小驼峰命名法。
* 类中控件名称必须与xml布局id保持一致。
* 用统一的量词通过在结尾处放置一个量词，就可创建更加统一的变量，它们更容易理解，也更容易搜索。
* 例如，请使用strCustomerFirst和strCustomerLast，而不要使用strFirstCustomer和strLastCustomer。

例：

mAdapter (类内变量）

##### 6.常量（Constants）

* 全部大写,采用下划线命名法.例如：MIN\_WIDTH

例：TYPE\_BROSWER

##### 7.资源文件（图片drawable文件夹下）

* 全部小写，采用下划线命名法，加前缀区分
* 命名模式：类型名称缩写逻辑名称/common逻辑名称
* 如果有多种形态如按钮等除外如btn\_xx.xml（selector）

例：

btn\_xx | 按钮图片使用btn\_整体效果（selector）

btn\_xx\_normal | 按钮图片使用btn\_正常情况效果

btn\_xx\_press | 按钮图片使用btn\_点击时候效果

bg\_head | 背景图片使用bg功能说明

def\_search\_cell | 默认图片使用def功能说明

icon\_more\_help | 图标图片使用icon功能说明

##### 8.资源布局文件（XML文件（layout布局文件））

* 全部小写，采用下划线命名法。

1. contentview命名, Activity默认布局，Fragment默认布局，不加任何后缀。 例：activity\_main.xml、fragment\_session.xml。

1. Dialog命名：dialog\_描述.xml 例：dialog\_normal.xml (Dialog命名)

1. PopupWindow命名：ppw\_描述.xml 例：pop\_info.xml (Dialog命名)

1. adapter的子布局：功能模块\_item.xml 例：item\_coupon.xml (item、cell的命名)

1. contentview中的布局。做为Activity或Fragment中的子布局，在contentview的命名后，跟上逻辑名称后缀。 例：activity\_main\_login.xml （意为：在main这个Activity中的一个login布局，若该布局可复用请删除"activiy\_"这段前缀）

##### 9.layout中的id命名

* 命名模式为：view缩写模块名称view的逻辑名称

特例:

Button | btn

* 除了以上这个特例，其他2个单词以上的一律取开头2个英文字幕做为大写。

例：ImageView | iv

完整例子：btn\_login、iv\_user

# iOS开发规范

## 一些代码规范

#### 禁止使用 xib 和 storyboard 进行页面设计。

#### 遵循 Apple 源代码的代码风格, 保持代码整洁优雅，我们是这样写 if 的程序员👩‍

#### if (somecode) {

}

#### 避免大量的 Magic number, 尽量使逻辑简单易懂

#### 推荐 工厂模式 适配器模式等 , 尽量解耦

#### 此模块未来私有库名称为 TAAIVideo 请依照此名字作为后续文件的命名规范 驼峰式命名

#### 模块将以私有库的形式集成, 所以建议以私有库的形式开发, 否则可能出现图片引用不到,文件路径不同等问题

#### 使用 OC 语言

#### 避免卡顿, 内存泄漏, 循环引用, 掉帧等

#### 通用命名规则

1. 自我描述性。属性*函数*参数*变量*常量/宏 的命名必须具有自我描述性。杜绝过度缩写、或者无意义的命名方式。

1. 驼峰命名方式。参数名、成员变量、局部变量、属性名都要采用小写字母开头的驼峰命名方式。

如果方法名以一个众所周知的大写缩略词开始，可以不适用驼峰命名方式。比如FTP、WWW等。

insertObject:atIndex: 规范的写法

insert:at: 不清晰，插入什么？at代表什么？

removeObjectAtIndex: 规范的写法

removeObject: 规范的写法，因为参数指明了要移除一个对象

remove: 不清晰，移除什么？

#### Category命名规范

1. Category中不要声明属性和成员变量。
2. 避免category中的方法覆盖系统方法。可以使用前缀来区分系统方法和category方法。但前缀不要仅仅使用下划线”\_“。

#### Block规范

1. 调用block时需要对block判空
2. 注意block潜在的循环引用

#### Notification

1. 避免通知满天飞，必须要思考，有没有更好的办法来代替这个通知，禁止遇到问题就想到通知。
2. 在多线程应用中，Notification在哪个线程中post，就在哪个线程中被转发，而不一定是在注册观察者的那个线程中。如果post消息不在主线程，而接受消息的回调里做了UI操作，需要让其在主线程执行。

#### 代码注释规范

1. .h 文件中暴露出的方法要加注释说明 （并表明副作用等）

#### UIViewController 代码示例

#pragma mark - Life Cycle

- (instancetype)init {}

- (void)dealloc {}

- (void)viewDidLoad {}

- (void)viewWillAppear:(BOOL)animated {}

#pragma mark - Fetch

- (void)fetchData {}

- (void)fetchMoreData{}

#pragma mark - Public

- (void)publicMethod {}

#pragma mark - Private

- (void)privateMethod {}

#pragma mark - Delegate

#pragma mark - UITableViewDataSource

#pragma mark - UITableViewDelegate

#pragma mark - OtherDelegate

#### 严禁超过 3 层的继承，试着思考使用协议，代理等方式

## 一些业务规范，技术选型

#### 视频播放部分目前了解下来是 Mp4 点播, 可以用系统 AVPlayer 进行封装，如果系统的可以满足需求避免使用第三方播放库

#### 工程全部为 Pod 管理, 如需引用新增第三方 Pod 请提前联系商讨, 尽量不再新增

#### 目前已经在用的 Pod 和版本（可以选择使用）

Installing AFNetworking (4.0.1)

Installing Aspects (1.4.1)

Installing KVOController (1.2.0)

Installing MJRefresh (3.4.1)

Installing Masonry (1.1.0)

Installing SAMKeychain (1.5.3)

Installing SDWebImage (5.6.1)

Installing TXLiteAVSDK\_Player (7.2.8927)

Installing WebViewJavascriptBridge (6.0.3)

Installing YYModel (1.0.4)

Installing YYText (1.0.7)

#### 希望最终以 Pod 的形式接入，尽量 “一行代码调用”，功能插件化， 即插即用

#### 为每一个接口创建单独的 Model 类，避免出现“上帝”Model， 用在所有地方

# iOS补充规范

## 一些代码规范

#### 禁止使用 xib 和 storyboard 进行页面设计。

#### 遵循 Apple 源代码的代码风格, 保持代码整洁优雅，我们是这样写 if 的程序员👩‍💻

**if** (somecode) {

}

#### 避免大量的 Magic number, 尽量使逻辑简单易懂

#### 推荐 工厂模式 适配器模式等 , 尽量解耦

#### 此模块未来私有库名称为 TAAIVideo 请依照此名字作为后续文件的命名规范 驼峰式命名

#### 模块将以私有库的形式集成, 所以建议以私有库的形式开发, 否则可能出现图片引用不到,文件路径不同等问题

#### 使用 OC 语言

#### 避免卡顿, 内存泄漏, 循环引用, 掉帧等

#### 通用命名规则

1. 自我描述性。属性函数参数变量常量/宏 的命名必须具有自我描述性。杜绝过度缩写、或者无意义的命名方式。
2. 驼峰命名方式。参数名、成员变量、局部变量、属性名都要采用小写字母开头的驼峰命名方式。 如果方法名以一个众所周知的大写缩略词开始，可以不适用驼峰命名方式。比如FTP、WWW等。

insertObject:atIndex: 规范的写法

insert:at: 不清晰，插入什么？at代表什么？

removeObjectAtIndex: 规范的写法

removeObject: 规范的写法，因为参数指明了要移除一个对象

remove: 不清晰，移除什么？

#### Category命名规范

1. Category中不要声明属性和成员变量。
2. 避免category中的方法覆盖系统方法。可以使用前缀来区分系统方法和category方法。但前缀不要仅仅使用下划线”\_“。

#### Block规范

1. 调用block时需要对block判空
2. 注意block潜在的循环引用

#### Notification

1. 避免通知满天飞，必须要思考，有没有更好的办法来代替这个通知，禁止遇到问题就想到通知。
2. 在多线程应用中，Notification在哪个线程中post，就在哪个线程中被转发，而不一定是在注册观察者的那个线程中。如果post消息不在主线程，而接受消息的回调里做了UI操作，需要让其在主线程执行。

#### 代码注释规范

1. .h 文件中暴露出的方法要加注释说明 （并表明副作用等）

#### UIViewController 代码示例

#**pragma** mark - Life Cycle

- (instancetype)init {}

- (**void**)dealloc {}

- (**void**)viewDidLoad {}

- (**void**)viewWillAppear:(BOOL)animated {}

#**pragma** mark - Fetch

- (**void**)fetchData {}

- (**void**)fetchMoreData{}

#**pragma** mark - Public

- (**void**)publicMethod {}

#**pragma** mark - Private

- (**void**)privateMethod {}

#**pragma** mark - Delegate

#**pragma** mark - UITableViewDataSource

#**pragma** mark - UITableViewDelegate

#**pragma** mark - OtherDelegate

#### 严禁超过 3 层的继承，试着思考使用协议，代理等方式

## 一些业务规范，技术选型

#### 视频播放部分目前了解下来是 Mp4 点播, 可以用系统 AVPlayer 进行封装，如果系统的可以满足需求避免使用第三方播放库

#### 工程全部为 Pod 管理, 如需引用新增第三方 Pod 请提前联系商讨, 尽量不再新增

#### 目前已经在用的 Pod 和版本（可以选择使用）

Installing AFNetworking (4.0.1)

Installing Aspects (1.4.1)

Installing KVOController (1.2.0)

Installing MJRefresh (3.4.1)

Installing Masonry (1.1.0)

Installing SA**MKeychain** (1.5.3)

Installing SDWebImage (5.6.1)

Installing TXLite**AVSDK\_Player** (7.2.8927)

Installing WebViewJavascriptBridge (6.0.3)

Installing YYModel (1.0.4)

Installing YYText (1.0.7)

#### 希望最终以 Pod 的形式接入，尽量 “一行代码调用”，功能插件化， 即插即用

#### 为每一个接口创建单独的 Model 类，避免出现“上帝”Model， 用在所有地方

## 补充新增

#### 系统最低支持 iOS 10， 使用 Xcode 11 以上版本

#### 由于 TutorABC 项目是 Universal 的，iPhone 仅支持竖屏，iPad 支持除 Upside Down 旋转， 所以代码 UI 要考虑到旋转的支持， 在相应旋转下做出丝滑的改变， 禁止使用监听旋转发通知的方式， 而应该使用系统自带的 api 如下

- (**void**)viewWillTransitionToSize:(**CGSize**)size

withTransitionCoordinator:(**id**<**UIViewControllerTransitionCoordinator**>)coordinator

另外需要注意的是，成人项目中通过 VC 的 Category 统一控制旋转，如需单独控制某个 VC 的旋转， 请复写下面的旋转相关方法

#pragma mark - Rotation

- (BOOL)shouldAutorotate {

}

- (UIInterfaceOrientationMask)supportedInterfaceOrientations {

}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation {

}

Java 后端开发版本规范：

1.后端框架及版本（JDK1.8、jar、springBoot2.0+），以vmd-aicourse工程基础进行编码

2.服务器操作系统、程序运行载体及版本（linux64bit、centOS7.3、nginx1.14、mysql5.6...）

3.组件或中间件版本（maven库地址（阿里云镜像）、logback/log4j、redis中间件cachecloud（iTG协助））

4.项目命名规范（管理接口：manage/\*\*; 对外业务接口：api/\*\* 详见vmd-aicourse工程）

5.接口域名+接口命名规范（调用第三方接口：rpc包下）

6.数据库设计及表、字段命名规范（库名：vmd\_business;表前缀：ai\_;字符格式：utf8mb4; 见补充“数据库规范.txt”）

7.项目中使用到的任何地址、时间（yyyy-MM-dd HH:mm:ss）等全部用统一配置

8.HTTP接口重试3次，超时时间（10s）设定