# 图腾泰科开发规范

1. 代码风格规范参考官方规范和样例
2. Java代码规范

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

1. Android代码规范

<http://source.android.com/source/code-style.html>

中文版本见《Code Conventions for Java》

1. iOS代码规范

《苹果 Cocoa 编码规范(中文版).pdf》

1. 变量命名要求遵循见名知义的原则，如果你的命名别人看不懂，或者过了2周自己也不知道这个变量是做什么的，那这个命名就失败鸟
2. 变量命名尽量少使用单词缩写
3. 大量的命名，要使用前缀和后缀的方式，会使命名看起来很统一

//订单状态

typedef NS\_ENUM(NSInteger, GHSOrderStatus)

{

GHSOrderStatusCanceled = 0, //已取消

GHSOrderStatusPendingPayment = 11, //待付款

GHSOrderStatusPendingDelivery = 20,//待发货

GHSOrderStatusDelivering = 30, //已发货

GHSOrderStatusReceived = 39, //已签收

GHSOrderStatusFinished = 40, //交易成功

GHSOrderStatusRejection = 63 //拒收

};

资源文件：

home\_logo.png

home\_item\_line.png

home\_icon\_search.png

home\_icon\_cancel.png

1. 养成使用常量和宏的好习惯，尽力避免使用magic number。如：

GHSPayTypeCashOnDelivery = 6, //货到付款

GHSPayTypeAlipay = 7 //支付宝

if(payType == 6) 这种代码别人是看不懂的

1. 业务逻辑尽量使用常量和宏，统一、见名知义、维护简单。

不要使用显示值来做业务逻辑，禁止下列行为：

~~if(status == “无货”)~~

~~{~~

~~disableBuyButton();~~

~~}~~

必须使用下列方式：

if( stock == 0 || status == STOCK\_EMPTY)

{

disableBuyButton();

}

1. 养成注释的好习惯，下列地方要求加注释
2. 业务逻辑复杂的地方
3. 变量需要释义和解释的地方
4. 暴露给别人掉用的接口
5. 如果通过变量和命名已经很清晰业务逻辑，注释就是画蛇添足，例如：

if ( firstLaunch )

{

gotoRegister();

}

1. 尽量少地暴露，变量作用域越小越好。
2. 能用本模块完成的，就不用全局变量。
3. 只在本模块用的变量，不要定义在头文件里
4. 只在1个方法用到的变量，不要写成类变量
5. 尽量少使用全局变量
6. 代码封装，实现统一、容易维护、容易理解
7. 方法实现写50行以上，考虑一下是否有必要封装成另外一个方法
8. 一段代码被拷贝了3次，那就应该考虑封装成1个方法
9. 一个代码缩进了5层以上，那最好单独写个方法调用
10. 尽量使用所见即所得编辑器制作界面，下列情形才考虑使用代码实现界面
11. 编辑器无法实现的界面
12. 复用性很强的界面
13. 界面制作使用恰当的占位图片和占位字符，尽量模仿真实数据。
14. 界面实现必须完全按照效果图来做，注意控件的坐标、大小尺寸，对齐、字体、颜色、阴影等
15. 界面上按钮可点击区域要大于44 x 44

在界面布局不允许这么大的情况下，尽量扩大可点击区域

1. 文本显示区域要尽可能地大，以避免文本显示不全

需要计算尺寸的地方，要进行计算然后调整相应的位置

1. 界面上所有接受按下事件的的按钮和区域，都需要有按下效果

Android按下效果使用公司封装的Highlight Layout组件

1. iOS需要准备@2x.png @3x.png资源文件，拉伸的地方要使用代码拉伸

Android尽量使用.9.png图片处理拉伸

1. iOS界面制作要求使用Auto Layout进行适配
2. Android界面适配策略

使用同1套layout文件，大于4.2寸的大屏幕，我们使用第2套dimens.xml，这个文件位于sw360dp目录下(意思就是宽度大于360dp的设备使用这套尺寸)，主要定义一些控件的宽高，间距，字体大小等等。可以在布局编辑器里，使用5寸左右的手机进行查看和修改

1. 尺寸使用dp，字体也是用dp，任何密度的屏幕上大小不变
2. 图片的宽高比必须保持一致，避免拉伸

可以固定写死宽高，也可以程序计算

1. 想清楚控件是固定宽度？固定高度？固定间距？还是固定宽高比
2. 想清楚控件是左对齐？中间对齐？还是右对齐？还是居中对齐
3. 字体，颜色，文本等放在资源文件中统一管理
4. 共用的组件比如导航栏、标题、字体样式等要写在同一个地方
5. 一般导航栏高度，列表的行高等使用固定数值
6. 固定尺寸时dp的数值确定（效果图1080x1920）

4.2寸及以下屏幕dp = px/3.37

4.2寸以上的屏幕dp = px/3，只有大屏幕效果不理想的情况下才用

1. 显示数据需要注意的地方
2. 视图加载完成后，清空占位数据，加载上个页面传过来的数据
3. 如果页面有缓存，需要加载缓存数据
4. 如果数据需要刷新，在视图将要显示时，根据刷新条件进行刷新
5. 显示数据建议封装成统一的方法比如showUser(User user)
6. 提交数据时
7. 需要使用Loading框或者把按钮置为不可用，以免重复提交
8. 需要对数据进行非空和格式校验
9. 需要按照返回的错误代码进行恰当的错误提示
10. 类库和解决方案选择的原则
11. 使用官方提供的方案
12. 使用图腾自己封装好的类库和工具，统一成熟、有人测试有人维护
13. 使用github和stackoverflow等网站评分较高的开源方案
14. 使用Google搜索出来靠前方案
15. 使用活跃的开源社区提供的方案
16. 使用比较新的方案
17. 实在找不到了才考虑自己实现
18. 使用公司提供的BaseActivity或TTBaseViewController，使用它提供的加载框，各种提示框，这样可以在一个项目中统一样式，也减少了工作量