1. 项目结构类

1.1

 包（packages）: 采用反域名命名规则，全部使用小写字母。一级包名为com，二级包名为xx（可以是公司或则个人的随便），三级包名根据应用进行命名，四级包名为模块名或层级名

|  |  |
| --- | --- |
| 包名 | 此包中含有 |
| com.XX.应用名.adapter | 界面用到的adapter类 |
| com.XX.应用名.tools | 公共工具方法类 |
| com.XX.应用名.base | 公共基类 |
| com.XX.应用名.bean | Json解析的实体类 |
| com.XX.应用名.view | 自定义view类 |
| com.XX.应用名.network | 网络交互类 |
| com.XX.应用名.service | Service组件 |

1.2

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。

|  |  |  |
| --- | --- | --- |
| 类 | 描述 | 例子 |
| Activity类 | Activity为后缀标识 | HomeActivity |
| Adapter类 | Adapter为后缀标识 | BaseAdapter |
| 公共方法 | Tool或者Util为后缀标识 | FileUtil、DateTool、ToastTool |
| Service类 | Service为后缀标识 | LogService |
| Fragment类 | Fragment为后缀标识 | CoursesFragment |
| 实体类 | Entity或者Info为后缀标识 | ActionInfo、MsgEntity |

1.3

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

如果有多种形态如按钮效果：btn\_xx\_selector.xml（selector）

|  |  |
| --- | --- |
| btn\_xx | 无效果 |
| btn\_xx\_normal | 正常效果 |
| btn\_xx\_press | 点击效果 |

Shape图形类：

命名模式：shape\_形状\_颜色

如：shape\_circle\_white

1.4

 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. 功能模块.xml 例如：activity\_course.xml、activity\_login.xml
2. Dialog命名 例如：dialog\_date\_list.xml
3. Popupwindow命名 例如：pop\_photo.xml
4. Fragment命名 例如：fragment\_fans.xml
5. 列表项命名 例如：item\_booking\_time.xml
6. Include项 例如：include\_head.xml
7. 子view项 例如：view\_dialog.xml
8. 控件、变量命名规则

2.1

控件id命名：

命名模式为：view缩写\_模块名称\_view的逻辑名称 例如：tv\_invitation\_choicetime

**（此部分不一定要完全按照此格式， 但必须标明该控件是什么，如textview可用tv或者text。但严格杜绝使用w1,w2,w3,text1,text2等格式来命名）**

控件的缩写参考表：

|  |  |
| --- | --- |
| 控件 | 缩写 |
| LayoutView | xx\_layout |
| RelativeView | rv |
| TextView | tv |
| Button | btn |
| ImageButton | Img\_btn |
| ImageView | img |
| EditText | edt\_txt |
| SeekBar | sk\_bar |
| listView | lv |

2.2

方法（methods）：动词或动名词，采用小驼峰命名法例如:onCreate(),run()

**（非必要完全按照此格式，但一定要能体现出方法的作用，不能用setdata1（）、setdata2（）等相似度高的命名，需要体现方法在类中的功能唯一性）**

2.3

变量（variables）：类中的控件变量名称尽量与xml布局中的id保持大体一致，提高辨认率。

2.4

常量（Constants）全部大写,采用下划线命名法.例如：MIN\_WIDTH

**（代码中尽量不要出现魔鬼数字，需要用数字时，请定义常量）**

1. 常见英文单词缩写（非必要，参考用）

|  |  |
| --- | --- |
| 名称 | 缩写 |
| average | avg |
| background | bg |
| buffer | buf |
| control | ctrl |
| delete | del |
| document | doc |
| error | err |
| escape | esc |
| increment | inc |
| infomation | info |
| initial | init |
| image | img |
| internationalization | I18N |
| length | len |
| library | lib |
| message | msg |
| password | pwd |
| position | pos |
| string | str |

1. Android编码规范建议

1.java代码中尽量不出现中文，最多注释中可以出现中文

2.局部变量命名、静态成员变量命名

只能包含字母，单词首字母除第一个外，都为大写，其他字母都为小写

3.常量命名

只能包含字母和\_，字母全部大写，单词之间用\_隔开

4.图片尽量分拆成多个可重用的图片

5.服务端可以实现的，就不要放在客户端

6.引用第三方库要慎重，避免应用大容量的第三方库，导致客户端包非常大

7.处理应用全局异常和错误，将错误以邮件的形式发送给服务端

8.图片的.9处理

9.使用静态变量方式实现界面间共享要慎重

10.Log(系统名称模块名称接口名称，详细描述)

11.不要重用父类的handler，对应一个类的handler也不应该让其子类用到，否则会导致message.what冲突

12.activity中在一个View.OnClickListener中处理所有的逻辑

13.strings.xml中使用%1$s实现字符串的通配

14.如果多个Activity中包含共同的UI处理，那么可以提炼一个CommonActivity，把通用部分叫由它来处理，其他activity只要继承它即可

15.数据一定要效验，例如

字符型转数字型，如果转换失败一定要有缺省值；

服务端响应数据是否有效判断