**Android命名规范**

**撰写人: \_\_桂明星\_\_ \_\_\_\_**

**日 期：\_\_ 2016年3月29日 \_\_**

**文档修订记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 修改日期 | 修改的章节 | 修改类型 | 修改描述 | 修改人 | 审核人 | 版本号 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* 修改类型分为 **A** – ADDED（增加） **M** – MODIFIED（修改） **D** – DELETED（删除）

修订说明：

1.文件修订时，应将关键的修订内容填入本表

2.对于文件整章节的大规模变动为大版本号变更，否则为小版本的增加

**\* 保密和变更申明**

这份文档包含了来自上海容瀚网络信息科技有限公司的可靠、权威的信息，这些信息是作为公司专用，文档的阅读者应对其内容保密，未经公司书面请求和书面认可，不得复制、泄露或散布这份文档。对这份文档内容的任何形式的泄露、复制或散布都是被禁止的。

目录

[一、 四种标识符命名法 3](#_Toc6279)

[二、 包名/类名命名 3](#_Toc31499)

[三、 方法命名 3](#_Toc1078)

[四、 变量命名 3](#_Toc28670)

[五、 常量命名 4](#_Toc2682)

[六、 图片文件命名 4](#_Toc1656)

[七、 布局文件命名 5](#_Toc16727)

[八、 布局文件ID命名 5](#_Toc4257)

[九、 strings文件 6](#_Toc15807)

[十、 color.xml 6](#_Toc21859)

[十一、 注释 6](#_Toc8621)

[十二、 APK打包命名 7](#_Toc29344)

## 四种标识符命名法

1. 驼峰(Camel)命名法:又称小驼峰命名法，除首单词外，其余所有单词的第一个字母大写。
2. 帕斯卡(pascal)命名法:又称大驼峰命名法，所有单词的第一个字母大写
3. 下划线命名法:单词与单词间用下划线做间隔。
4. 匈牙利命名法:广泛应用于微软编程环境中，在以Pascal命名法的变量前附加小写序列说明该变量的类型。 量的取名方式为：<scope\_> + <prefix\_> + <qualifier>范围前缀，类型前缀，限定词。

## 包名/类名命名

包（packages）: 全部使用小写字母。一级包名为com，二级包名为ifuifu，三级包名根据应用进行命名，四级包名为模块名或层级名，五级为子模块名称。比如医数病人端名称简写为（customer）。那么包名则为com.ifuifu.customer

|  |  |
| --- | --- |
| **包名** | **包中类名** |
| com.ifuifu.xxxxx.activities | 包中文件名XxxActivity，Xxx为页面名称如（LoginActivity） |
| com.ifuifu.xxxxx.activities.login | Activity下的子模块分类 |
| Com.ifuifu.xxxxx.activities.login.veiw | 自定义View类,文件名为XxxView,一般与业务逻辑紧密相关封装提取的视图 |
| com.ifuifu.xxxxx.base | 基础类(BaseActiviy,BaseFragment等) |
| com.ifuifu.xxxxx.adapter | 包中文件名XxxAdapter |
| com.ifuifu.xxxxx.utils | 工具类,文件名为XxxUtil |
| com.ifuifu.xxxxx.beans | 实体类 |
| com.ifuifu.xxxxx.beans.to | 实体类下，请求参数实体，命名：XxxEntity |
| com.ifuifu.xxxxx.beans.vo | 实体类下，解析数据实体，命名：XxxBean |
| com.ifuifu.xxxxx.db | 数据库操作类 |
| com.ifuifu.xxxxx.manager | 模块管理类,文件名为XManager |
| com.ifuifu.xxxxx.fragments | 碎片,文件名为XxxFragment |
| com.ifuifu.xxxxx.imps | 接口,文件名为XxxImp |
| com.ifuifu.xxxxx.widget | 自定义控件类，与业务逻辑无关组件可供外部类直接引用 |
| com.ifuifu.xxxxx.service | Service服务,文件名为XxxService |
| com.ifuifu.xxxxx.constants | 常量类，文件名为XxxConstant,可放枚举常量EnumConstants，HttpConstants请求地址常量等 |

## 方法命名

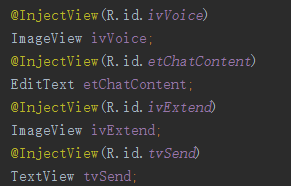
采用小驼峰命名法,例如setData()，initView();

## 变量命名

变量命名采用小驼峰命名法,类中控件名称必须与xml布局id保持一致;

**示例如下:**

a.注解方式：



b.普通方式：

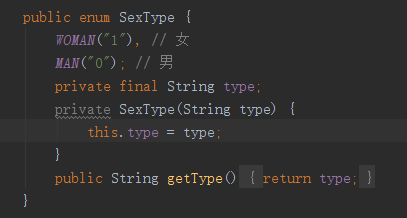
private Button testInsertOne;  
private Button testInsertList;  
private Button testQueryOne;  
private void assignViews() {  
 testInsertOne = (Button) findViewById(R.id.*testInsertOne*);  
 testInsertList = (Button) findViewById(R.id.*testInsertList*);  
 testQueryOne = (Button) findViewById(R.id.*testQueryOne*);  
}

## 常量命名

全部大写,采用下划线命名法.例如: public final static String INTENT\_DATA;

枚举enum 类型，因为他们是常量，所以枚举类型的字段的名字一般是大写字母

例如：



## 图片文件命名

全部小写,采用下划线命名法,加前缀区分

命名模式:控件\_逻辑名称/功能\_逻辑名称.

|  |  |
| --- | --- |
| 名称 | 功能 |
| btn\_xx\_normal.png | 按钮图片正常效果 |
| btn\_xx\_pressed.png | 按钮图片按下效果 |
| btn\_xx\_selector.xml(处理多张图片显示) | 按钮图片选择器 |
| btn\_xx\_disable.png | 按钮图片禁用效果 |
| btn\_xx\_checked.png | 按钮checkbox图片选择效果 |
| btn\_xx\_uncheck.png | 按钮checkbox图片未选中效果 |
| repeat\_xxx.xml(tileMode图片命名) | 图片复制效果文件命名 |
| def\_xx\_bg.png | 默认图片文件命名 |
| ic\_laucher.png | 默认是app应用logo图标直接替换系统图标 |

**不同分辨率对应资源**

三种不同的密度：高（ hdpi、xhdpi、xxhdpi），中（ mdpi）和低（ ldpi）

|  |  |  |
| --- | --- | --- |
| 资源目录 | ic\_launcher.png图标大小 | 分辨率 |
| drawable-hdpi | 72\*72 | 4:3（例如：480\*320） |
| 5:3（例如：800\*480） |
| drawable-mdpi | 48\*48 | 16:9（例如：960\*540） |
| drawable-xhdpi | 96\*96 | 1280\*720  标清 |
| drawable-xxhdpi | 144\*144 | 1920\*1080 高清 |

注：发布到googleplay市场，app程序logo大小是512\*512

Selector代码示例:

<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android" >  
 <item android:state\_pressed="true" android:drawable="@drawable/add\_medicine\_pressed" />  
 <item android:drawable="@drawable/add\_medicine\_normal" />  
</selector>

tileMode代码示例:

<?xml version="1.0" encoding="utf-8"?>  
<bitmap xmlns:android="http://schemas.android.com/apk/res/android"  
 android:src="@drawable/chat\_dotted\_line"  
 android:tileMode="repeat" />

## 布局文件命名

全部小写,采用下划线命名法

1. **Activity布局命名**

activity\_功能模块.xml例如:activity\_login.xml

1. **Dialog布局命名**

dialog\_功能模块.xml 例如:dialog\_common.xml

1. **PopupWindow命名**

pop\_功能模块.xml 例如:pop\_info.xml

1. **ListItem命名**

listitem\_功能模块.xml 例如:listitem\_city.xml

1. **include模块命名**

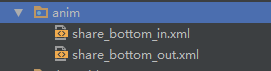
include\_功能模块.xml 例如:include\_city.xml

1. **View自定义命名**

view\_功能模块.xml 例如:view\_search.xml

## Anim动画资源命名

全部单词小写，单词间以下划线分割，命名规则：模块名\_逻辑名称

如：

## 布局文件ID命名

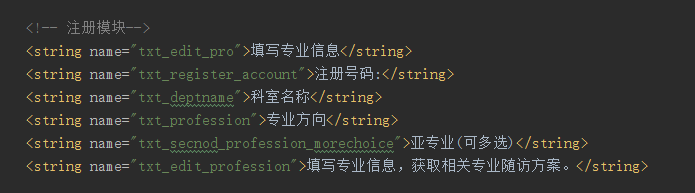
|  |  |
| --- | --- |
| 控件 | 缩写 |
| ImageView | iv |
| TextView | tv |
| Button | btn |
| ImageButton | ib |
| CheckBox | cb |
| RadioButton | rb |
| DigtalClock | dc |
| DatePicker | dp |
| EditText | et |
| TimePicker | tp |
| ProgressBar | pb |
| SeekBar | sb |
| WebView | wv |
| Tab | tab |
| Spinner | spi |
| ScrollView | sv |
| ListView | lv |
| ExpandableList | expL |
| LinearLayout | ll |
| FramLayout | fl |
| RelativeLayout | rl |

## strings文件

1. Java代码中不出现中文,最多注释中可以出现中文,便于后期国际化
2. 在values文件下strings里命名按功能模块划分，模块上加中文注释<!--xxxxxx-->

显示文字命名：txt\_模块名\_逻辑名称;

Toast提示文字命名：toast\_模块名\_逻辑名称;

1. 

## color.xml

通用的颜色色值放到color.xml文件中

通用色值配置示例:

<color name="bg\_c1">#F0F0F0</color> *<!-- 页面背景色调 -->*<color name="txt\_c2">#0189D0</color> *<!-- 文字颜色 -->*<color name="btn\_c3">#0192A1</color> *<!--按钮背景颜色 -->*<color name="c4">#FFFFFF</color> *<!-- 其它常规颜色 -->*

## 注释

1. 每个类名必须有注释,至少包含类描述,作者,联系方式

示例如下:

*/\*\*  
 \* 登录页面  
 \** ***@Author:*** *star*

*\** ***@Email:*** *guimingxing@163.com  
 \** ***@Date:*** *2015/7/3 0003  
 \*/*public class LoginActivity extends BaseActivity{

每个方法名也需要加注释,至少包括方法描述,参数描述

示例如下:

*/\*\*  
 \* 提交评论信息  
 \** ***@param*** *datas 数据  
 \** ***@param*** *curNo 评论编号  
 \** ***@param*** *cardNo 卡号  
 \** ***@param*** *content 评论内容  
 \*/*public void submitComments(String datas, String curNo, String cardNo, String content) {

## APK打包命名

应用发布apk版本命名规则：ifu+android+客户端名+版本号+渠道名.apk

例如：医数项目ifu\_android\_customer\_V2.24\_huawei.apk;