**深圳xxx有限公司**

**Android客户端代码规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **文档名称** | 客户端代码规范 | | |
| **项目名称** |  | | |
| **文件编号** |  | **版本** | V1.0 |
| **编写人** |  | **编写日期** |  |
| **审核** |  | **确认** |  |

**版本历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **起止日期** | **备注** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 一．编写目的

统一的编码规范可以使程序可读性增强，使程序更易于被理解，并降低维护成本。

## 二．预期读者及阅读建议

本规范文档的主要读者为Android开发团队成员。建议每位Android开发人员在进行代码编写前仔细阅读此文档，并按文档的规范要求进行编码。

## 三．Android编码规范

### 3.1命名规范

命名原则：

1. 使用完整的英文名进行命名，慎用字母缩写，如果使用，请按照缩写规范使用

例：No代表number、ID代表identification

1. 避免命名过长（20个字符内比较好）
2. 避免相似的命名

例：shopCar 和 shopCAR

eluxue命名规范：

1. 包（packages）: 采用反域名命名规范，全部使用小写字母。一级包名为net，二级包名为eluxue，三级包名根据应用进行命名，四级包名根据类的具体功能类型命名。

|  |  |  |
| --- | --- | --- |
| **包名** | **说明** |  |
| net.eluxue.应用.activity | 此包中包含：页面用到的Activity类 |  |
| net.eluxue.应用.handler | 此包中包含：解析服务器端返回信息的Handler类 |  |
| net.eluxue.应用.bean | 此包中包含：程序中要用到的JavaBean类 |  |
| net.eluxue.应用.adapter | 此包中包含：进行页面的数据适配的Adapter类 |  |
| net.eluxue.应用.info | 此包中包含：程序中要用到的非JavaBean的其它信息类， 例如：Constants.java |  |
| net.eluxue.应用.tools | 此包中包含：公共工具方法类 |  |
| net.eluxue.应用.db | 此包中包含：数据库操作类 |  |
| net.eluxue.应用.custom | 此包中包含：自定义的view等类 |  |
| net.eluxue.应用.interfaces | 此包中包含：所有接口 |  |
| net.eluxue.应用.service | 此包中包含：Service服务 |  |
| net.eluxue.应用.broadcast | 此包中包含：Broadcast服务 |  |

1. 类（classes）：名词，尽量避免缩写。首字母大写，非首字母全部小写。类名中包括多个单词时，采用驼峰标识方式。

|  |  |  |
| --- | --- | --- |
| **类名** | **规范写法** | **备注** |
| Activity类 | Aty后缀标识。欢迎页面类：WelcomeAty |  |
| Adapter类 | Adp后缀标识。商品详情适配器类：ProductDetailAdp |  |
| Bean类 | Bean后缀标识。安卓越Bean命名：AnjoyoPopBean |  |
| Constants类 | Constants后缀标识。 |  |
| 解析类 | Hlr后缀标识。首页解析类：HomePosterHlr |  |
| 公共方法类 | Tools或Manager后缀标识。  线程池管理类：ThreadPoolManager  日志工具类：LogTools |  |
| 数据库类 | DBHelper后缀标识。  城市数据库：CityDBHelper |  |
| 自定义View类 | 以功能名字成结尾：CustomListView |  |
| Service类 | Service后缀标识 |  |
| BroadcastReceive类 | Broadcast后缀标识 |  |

1. 接口（interface）：命名规则与类的命名规则一致。Able、er或ible结尾。
2. 方法（methods）：动词或动名词。第一个单词首字母小写。方法名中包括多个单词时，第一个单词之后的所有单词采用驼峰标识方式。

例如：onCreate()

run()

|  |  |  |
| --- | --- | --- |
| **方法名** | **规范写法** | **备注** |
| initXXX() | 初始化相关方法使用init前缀标识 |  |
| adapterXXX() | 适配相关使用adapter前缀标识 |  |
| isXXX() checkXXX() | 方法返回值为boolean型的请使用is或check前缀标识 |  |
| getXXX() | 返回值为基本类型或自定义类型的，使用get前缀标识 |  |
| processXXX() | 对数据进行处理的，尽量使用process前缀标识 |  |
| resetXXX() | 对数据重组的，使用reset前缀标识 |  |
| displayXXX() | 弹出提示框和提示信息，使用display前缀标识 |  |
| saveXXX() | 与保存数据相关的，使用save前缀标识 |  |
| clearXXX()removeXXX() | 清除数据相关的，使用clear或remove前缀标识 |  |
| drawXXX() | 绘制数据或效果相关的，使用draw前缀标识 |  |

1. 变量（variables）大小写规范与方法名一致。类中控件名称必须与xml布局id保持一致。

例如：filePath、fileName、textTitle、editUsername

1. 常量（constants）全部大写，单词之间用下划线隔开。

例如：MIN\_WIDTH

1. XML文件（布局文件）：统一用小写加下划线组合
2. Activity默认布局，以去掉后缀的Activity类进行命名。不加后缀。
3. Activity子布局，父布局名称开始，追加item 和子布局功能说明。

例如：Activity默认布局：homeposter.xml

子布局为 homeposter\_item\_poster.xml

1. 资源文件（图片）：统一用小写加下划线组合，加前缀区分。

|  |  |  |
| --- | --- | --- |
| **图片命名例子** | **规范写法** | **备注** |
| btn\_login\_normal | 按钮图片使用btn\_功能\_说明 |  |
| 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 |  |
| seg\_list\_line | 具有分隔特征的图片使用seg\_功能\_说明 |  |
| sel\_ok | 选择图标使用sel\_功能\_说明 |  |

1. 动画文件（anim包）：统一用小写加下划线组合，加前缀区分。

|  |  |  |
| --- | --- | --- |
| **动画命名例子** | **规范写法** | **备注** |
| click\_head\_left | 点击背景切换动画使用click前缀标识 |  |
| bg\_shape\_rectangle | 背景自定义图形使用bg前缀标识 |  |
| show\_shopcar\_add | 小动画效果使用show前缀标识 |  |

1. 资源ID（resources id）：大小写规范与方法名一致。命名规范为“资源控件的缩写名”+“变量名”。注意：页面控件名称应该和控件id名保持一致

|  |  |  |
| --- | --- | --- |
| **变量命名例子** | **规范写法** | **备注** |
| ImageView | imgXXX |  |
| TextView | textXXX |  |
| EditText | editXXX |  |
| Button | btnXXX |  |
| LinearLayout | linXXX |  |
| RelativeLayout | relXXX |  |
| FrameLayout | fraXXX |  |
| ScrollView | scrXXX |  |
| ListView | listXXX |  |
| GridView | gridXXX |  |
| Gallery | galXXX |  |
| Spinner | spinnerXXX |  |
| Custom | customXXX | 自定义控件 |

11、项目编码必须为“utf-8”。

### 3.2JAVA代码规范

1. 格式上层级间要用TAB制表符缩进
2. 条件分支要写清注释

错误样例：

**if**(errorState == 1){

thread();

}**elseif**(errorState == 2){

正确样例：

// 判断错误状态

**if**(errorState == 1){

// 向服务器端发送数据请求

thread();

}**elseif**(errorState == 2){

// 2…

1. 程序中无效的//TODO注释不应存在,做标识用的// TODO可以存在。
2. 程序中无效的引用不应存在
3. 尽量避免创建新的对象

错误样例：

用户输入：input

StringtempString = input;

StringfinalString = tempString.subString(1,3);

正确样例：

用户输入：input

StringfinalString = input.subString(1,3);

1. 尽可能使用本地方法：例如String.indexOf()，运行速度快
2. 尽量避免使用getter和setter，原因是虚方法的访问速度比直接访问慢
3. 使用实类比接口要好

错误样例：

Map map = new HashMap();

正确样例：

HashMap map = new HashMap();

1. 将成员变量缓存到本地进行操作

错误样例：

for(inti=0;i<10;i++) {

dumpItem(this.item[i]);

}

正确样例：

Item[] tempItem = this.item;

for(inti=0; i<10; i++) {

dumpItem(tempItem[i]);

}

1. 永远不要在for循环第二个参数中调用任何方法

错误样例：

for(inti=0; i<this.getCount(); i++) {

…

}

正确样例：

intcount = this.getCount();

for(inti=0; i<count; i++) {

…

}

1. 避免使用枚举类型，原因是会大幅度降低速度，并增加文件体积。
2. 条件控制语句、循环语句的嵌套使用尽量避免超过3层，特殊情况除外。

### 3.3XML布局代码规范

1. Layout标签使用长标签
2. 非Layout标签使用短标签
3. 层次间用TAB制表符缩进
4. 如没有特殊情况，一律使用dip和sp单位，代码中使用px，一律进行dip转化

### 3.4 代码注释规范

1. 文件注释
2. 所有的源文件都应该在开头有一个注释，其中列出类名、版本信息、日期和版权声明。

如下：

/\*

\* 文件名

\* 包含类名列表

\* 版本信息，版本号

\* 创建日期。

\* 版权声明

\*/

1. 类注释使用文档注释，要写明类的中文名、作者及版本

正确样例：

/\*\*

\* 类名

\* @author 作者 <br/>

\* 实现的主要功能。

\* 创建日期

\* 修改者，修改日期，修改内容。

\*/

Class IndexActivity {…

1. 类属性注释说明

**private**及**默认**访问标识修饰的类属性使用**单行注释**

**public**及**protected**访问标识修饰的类属性使用**文档注释**

正确样例：

private int dialogState = 0;// 进度框状态（0不显示，1显示）

int netOkFlag = 1;// 网络可用标识（0不可用，1可用）

/\*\*进度框状态（0不显示，1显示）\*/

public int dialogState = 0;

/\*\*网络可用标识（0不可用，1可用）\*/

protected int netOkFlag = 1;

1. 方法使用文档注释，对于有参数列表、返回值及抛出异常的方法，要有注释说明

正确样例：

/\*\*

\* 方法描述

\* @param参数描述

\* @return 返回值描述

\* @exception 可能抛出的异常

\*/

public Boolean drawTemp(num) throws IOException{

1. 当版本确定后，因业务变更而需**增加代码**时，要添加说明的注释

正确样例：

// added by 张三 at 2011-5-11begin

新增代码行

// added by 张三 at 2011-5-12 end

1. 当版本确定后，因业务变更而需**修改代码**时，要添加说明的注释

正确样例：

// updated by 李四 at 2011-5-11begin

修改代码行

//updated by 李四 at 2011-5-12 end

1. 当版本确定后，因业务变更而需**删除代码**时，要添加说明的注释

正确样例：

// deleted by 王五 at 2011-5-11begin

// 删除代码行

//deleted by 王五 at 2011-5-12 end