**Pull Bears技术文档**

－**关于Android编程规范的建议**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 更新日期 | 更新版本 | 更新人 | 更新内容 | 审批人 |
| 2015/8/6 | 1.0.0 | 陶昊 | 新建 |  |
|  |  |  |  |  |

目录

文档说明 3

背景介绍 3

编写目的： 3

适用范围： 3

文档内容 4

Eclipse文档结构 4

命名 7

类命名 7

方法的命名 7

变量命名 7

常量命名 8

layout命名 8

id命名 8

资源命名 8

注释 9

类注释 9

方法注释 9

其他注释 9

XML注释 9

代码风格 10

缩进 10

空行 10

参数和返回值 10

控制语句 10

异常的捕捉处理 11

括号 12

常用第三方类库推荐 13

HttpClient： 13

KSOAP2： 13

ZXing： 13

LeanCloud： 13

ShareSDK： 13

UMengSDK: 13

# 文档说明

## 背景介绍

Android分为四层：应用程序层（Applications）、应用程序框架层（Application Framework）、系统运行库层（Libraries和Android Runtime）和Linux内核层（Linux Kernel）。然而在我们一般的Android开发中指的是应用程序层开发。所有的应用程序都是使用Java语言编写的，通过调用应用程序框架层（Application Framework）所提供的API来完成。

## 编写目的：

一个软件的生命周期中，80%的花费在于维护。几乎没有任何一个软件，在其整个生命周期中，均由最初的开发人员来维护。编码规范可以改善软件的可读性，可以让程序员尽快而彻底地理解新的代码。

## 适用范围：

本文档描述的内容适用于使用Java作为开发语言开发Android应用或需要相关编程借鉴的相关人员。

# 文档内容

## Eclipse文档结构

a）在Root（如无特殊说明，Root在本文档代表在根目录下与工程同名文件夹）下建立libs文件夹，用于存放第三方类库。

b）在Root下建立src目录，用于存放Java类，每个Java类都是都在对应的包（package）里。

1. 一个唯一包名的前缀总是全部小写的ASCII 字母并且是一个顶级域名，通常是com，包命的后续部分根据项目的不同规范不尽相同。

2. 规约：比如项目是android demo，那么包命名必须以com.androiddemo开始，后面为模块名或层级名称。

3. com.androiddemo包名下面自定义一个MyApplication类用来存储一些系统的信息和初始化APP。

4. 通常根据模块名或层级名称还分utils（工具类）、activity（展现视图的controller层）、adapter（适配器）、entity（model层）、network（调用接口访问网络）、view（自定义控件）等。

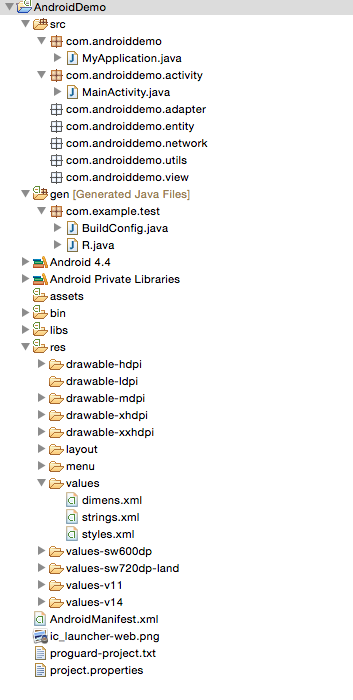
c）在Root下建立gen目录，该目录下面的包中主要有一个R.java文件。该文件是系统自动修改生成，不允许开发人员去手动修改。

d）在Root下建立res目录，res目录下面又有以下常用文件夹：

1. drawable：主要是存放所使用的图片。

2. layout：布局,这个就是经常看到的与用户交互的界面的xml 文件。

3. values: 是用程序人员设置的一些常量的XML文件，便于修改和使用，一般有colors ，dimens，strings，styles等。

e）在Root下建立AndroidManifest.xml文件，用于整个Android应用的配置。命名

## 类命名

类名必须使用驼峰规则，即首字母必须大写，如果为词组，则每个单词的首字母也必须要大写，类名必须使用名词，或名词词组。要求类名简单，不允许出现无意义的单词（如 class XXXActivity）。

## 方法的命名

命名规则：方法名是一个动词，采用大小写混合的方式，第一个单词的首字母小写，其后单词的首字母大写。

例如： public void close(); public String getUserName();

## 变量命名

第一个单词的首字母小写，其后单词的首字母大写。变量名不应以符号开头，尽管这在语法上是允许的。变量名应简短且富于描述。变量名的选用应该易于记忆。

## 常量命名

类常量的声明，应该全部大写，单词间用下划线隔开。

例如：static final int MIN\_WIDTH = 10;

## layout命名

layout xml 的命名必须以全部单词小写，单词间以下划线分割，并且使用名词或名词词组，即使用 模块名\_功能名称来命名。

如：activity\_main.xml

## id命名

layout中所使用的id必须以全部单词小写，单词间以下划线分割，并且使用名词或名词词组，并且要求能够通过id直接理解当前组件要实现的功能。

## 资源命名

layout中所使用的所有资源（如drawable,style等）命名必须以全部单词小写，单词间以下划线分割，并且尽可能的使用名词或名词组，即使用 模块名\_用途 来命名。如果为公共资源，如分割线等，则直接用用途来命名。

# 注释

## 类注释

每一个类都要包含如下格式的注释，以说明当前类的功能等。

/\*\*

\* 类名

\* @author 作者

\* 实现的主要功能。

\* 创建日期

\* 修改者，修改日期，修改内容。

\*/

## 方法注释

/\*\*

\* 方法的一句话概述

\* <p>方法详述（简单方法可不必详述）</p>

\* @param 说明参数含义

\* @return 说明返回值含义

\* @throws IOException 说明发生此异常的条件

\* @throws NullPointerException 说明发生此异常的条件

\*/

## 其他注释

如果需要多行 使用/\*。。。。 \*/形式，如果为单行是用//。。。。形式的注释。

## XML注释

使用<!-- -->注释，要求注释清晰易懂。

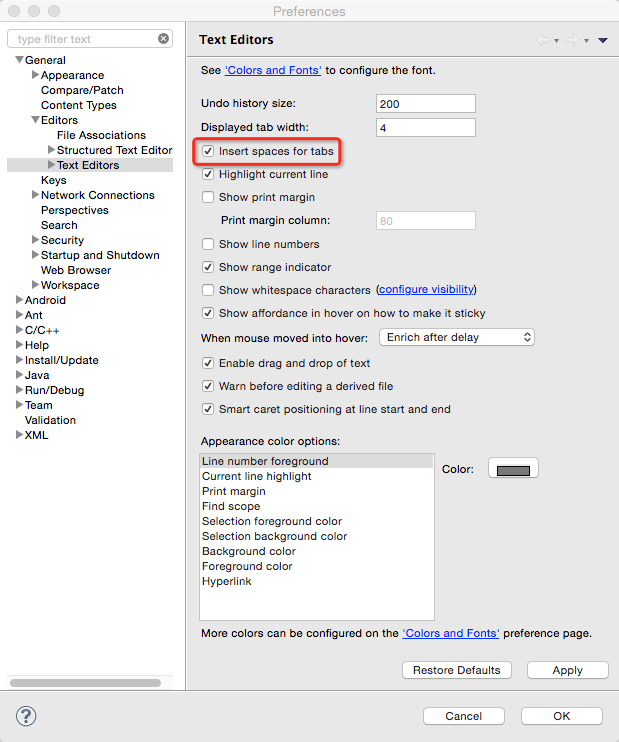
# 代码风格

## 缩进

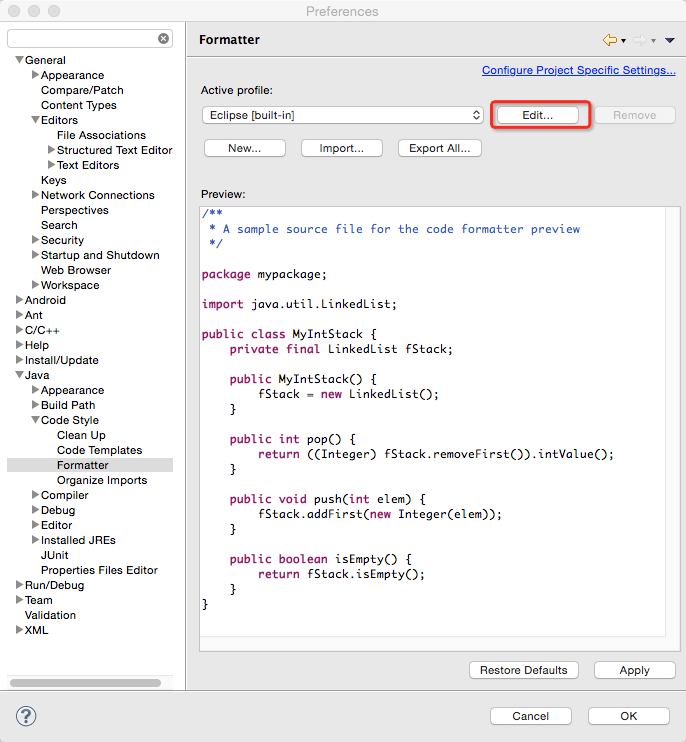
使用空格进行缩进，推荐缩进为4个空格。

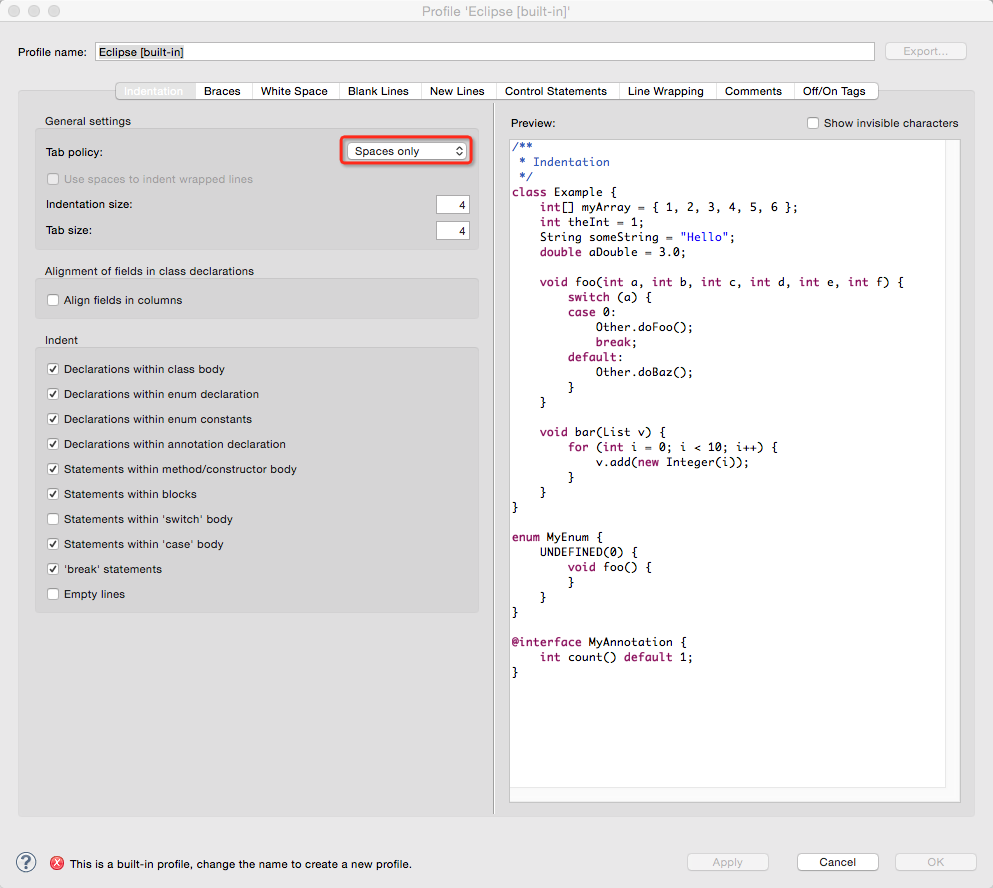
在eclipse中如果使用Tab键缩进需要进行设置。

如果是Windows系统：点击window-> preference；如果是Mac系统：点击ADT->偏好设置。然后依次选择 General->Editors->Text Editors,选中右侧的 insert space for tabs;如下图所示，保存，第一步完成；



然后再次点开preference ,依次选择 java（或C++）->code style ->formatter,点击右侧的editor，选则左侧 tab policy的值为spaces only,确定，应用保存即可，如下图所示：





## 空行

空行将逻辑相关的代码段分隔开，以提高可读性。

下列情况应该总是使用空行：

1. 两类变量之间。
2. 两个方法之间。
3. 方法内的局部变量和方法的第一条语句之间。
4. 一个方法内的两个逻辑段之间，用以提高可读性。

## 参数和返回值

1.一个方法的参数尽可能的不要超过4个。

2. 如果一个方法返回的是一个错误码，请使用异常。

## 控制语句

1． 判断中如有常量，则应将常量置于判断式的左侧。

如： if (“张三”.equals(getUserName()))...

2．所有if 语句必须用{}包括起来,即便是只有一句：

if (true){

//do something......

}

if (true)

i = 0; //不要使用这种

3. while(index < products.getCount()){

//每此都会执行一次getCount()方法，

//若此方法耗时则会影响执行效率

//而且可能带来同步问题，若有同步需求，请使用同步块或同步方法

}

//将操作结构保存在临时变量里，减少方法调用次数

final int count = products.getCount();

while(index < count){ }

## 异常的捕捉处理

1. 通常的思想是只对错误采用异常处理：逻辑和编程错误，设置错误，被破坏的数据，资源耗尽，等等。
2. 通常的法则是系统在正常状态下以及无重载和硬件失效状态下，不应产生任何异常。
3. 最小化从一个给定的抽象类中导出的异常的个数。对于经常发生的可预计事件不要采用异常。不要使用异常实现控制结构。
4. 若有finally 子句，则不要在try 块中直接返回，亦不要在finally 中直接返回。

## 括号

1. 圆括号：

一般而言，在含有多种运算符的表达式中使用圆括号来避免运算符优先级问题，是个好方法。

即使运算符的优先级对你而言可能很清楚，但对其他人未必如此。你不能假设别的程序员和你一样清楚运算符的优先级。 if (a == b && c == d) // 错误!

if ((a == b) && (c == d)) // 正确!

1. 大括号：

“{}”中“{”总是跟在类、方法、控制语句声明后面，不另起一行。

# 常用第三方类库推荐

### HttpClient：

完善的HTTP客户端，它提供了对HTTP协议的全面支持，可以使用HTTP GET和POST进行访问。

### KSOAP2：

调用WebService。

### ZXing：

实现二维码生成和解析。

### LeanCloud：

Android消息推送开发、实时通信服务。

### ShareSDK：

社会化组件，分享、分析等功能集成开发，也可采用友盟社会化组件。

### UMengSDK:

APP错误、用户行为等统计SDK。