|  |
| --- |
| 58同城信息技术有限公司 |
| 安卓开发代码规范 |
| V1.1 |
|  |
| **李金城** |
| **2014/7/28** |
| **周德山** |
| **2014/12/24** |

|  |
| --- |
| [更新记录 2](#_Toc5691)  [1. 文档描述 2](#_Toc10038)  [2. 业务线定义 3](#_Toc29619)  [3. 规范要求 3](#_Toc23076)  [3.1. Java代码规范 3](#_Toc9347)  [3.1.1包 3](#_Toc26622)  [3.1.2 类 4](#_Toc8889)  [3.1.3 接口 4](#_Toc23604)  [3.1.4 Import(插件实现) 4](#_Toc18227)  [3.1.5 全局变量 5](#_Toc660)  [3.1.6 方法 5](#_Toc21320)  [3.1.7 局部变量 6](#_Toc12499)  [3.1.8 注释 6](#_Toc24749)  [3.1.9 语句书写要求 7](#_Toc4445)  [3.2. 资源文件规范（这部分需要记忆） 8](#_Toc12670)  [3.2.1. Layout中的要求 8](#_Toc8636)  [3.2.2. strings.xml中的要求  9](#_Toc27026)  [3.2.3. drawable中的要求 9](#_Toc26895)  [4. 辅助工具 10](#_Toc17291)  [4.1. Code Templates 10](#_Toc20331)  [4.1.1. 导入统一模板 10](#_Toc10046)  [4.1.2. 修改模板配置 11](#_Toc1312)  [4.1.3. 验证 11](#_Toc29787)  [4.2. Formatter 12](#_Toc8496)  [4.2.1. 安装 13](#_Toc11153)  [4.2.2. 使用 13](#_Toc6271)  [4.3. Check Style 13](#_Toc12662)  [4.3.1. 安装 14](#_Toc31619)  [4.3.2. 配置 14](#_Toc501)  [4.3.3. 使用 15](#_Toc5761)  [4.4. FindBugs 16](#_Toc8228)  [4.4.1. 安装 16](#_Toc12439)  [4.4.2. 配置 17](#_Toc11676)  [4.4.3. FindBugs使用 18](#_Toc18462)  [4.4.4.导入android studio的setting文件 18](#_Toc11732) |

更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **更新内容** | **更新时间** | **更新者** |
| **1.0** | 创建初稿 | 2014.7.28 | 李金城 |
| **1.1** | 添加注释要求、添加通用业务定义、修改Java变量规范、增加laout规范、drawable规范、String规范、注释规范 | 2014.7.29 | 李金城 |
| **2.0** | 增加JAVA代码的通用规范，包括包，类，代码导入，全局变量，局部变量，方法，语句，注释 | 2014.12.24 | 周德山 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 文档描述

该文档规范了IM前端编码时重点注意事项，前端开发维护人员必须认真阅读、理解和严格遵守该规范。

制定原则：

1.简明扼要，容易记忆，方便日后推行

2.达到业务线间文件容易区分，代码风格80%相似

3.结构清晰，方便日后查找

# 业务线定义

|  |  |  |
| --- | --- | --- |
| **业务线** | **描述** | **缩写** |
| **登陆业务**  **login** | 帮帮登陆相关业务总称。包括：登陆58、获取ppu等 | login |
| **聊天沟通业务**  **Instant messaging** | 采用TCP方式与服务器通信的全部业务总称。包括：登陆IM、拉取好友、拉取分组、拉取离线、收发消息、系统消息、系统通知、足迹、推荐等 | im |
| **在线收入业务**  **Online income** | 58在线广告相关业务总称。包括：精准、置顶、刷新、智能等 | income |
| **房产业务**  **House property** | 房产相关功能总称。包括：房产工作台、信息列表、详情页等 | house |
| **招聘业务**  **job** | 招聘相关功能总称。包括：消息流、简历管理、职位管理、人才优选、简历搜索等 | job |
| **二手车业务**  **second-hand car** | 二手车相关功能总称。包括：二手车工作台、信息列表、详情页等 | car |
| **生活服务（黄页）业务**  **Life service** | 黄页相关功能总称。包括：消息流工作台、订单管理、评价管理等 | life |
| **信息发布业务**  **publish** | 信息发布相关业务总称。包括：房产发布、招聘发布职位、二手车发布等 | publish |
| **通用业务**  **Common** | 通用组件、通用对话框、通用页面等 | common |

# 规范要求

## Java代码规范

程序中**不允许使用拼音**不熟悉的单词，在线翻译选择能够表达意思的最常用单词

### 3.1.1包

|  |  |
| --- | --- |
| 要求 | 1.**全部英文小写**  2.包名固定以：**com.wuba.bangbang 开头** |
| 正例 | com.bangbang.wuba.login |
| 反例 | com.bangbang.login  com.bangbang.wuba.Login |
| 控制方式 | 人为控制 |

### 3.1.2 类

|  |  |
| --- | --- |
| 要求 | 1.类名都以**UpperCamelCase**风格编写  2.Activity命名**以Activity作为后缀**  3.**缩写词**字母全部**大写**  4.类名要用名词  5.抽象类需要Abs开头 |
| 正例 | ChatActivity, IMSocket |
| 反例 | chatActivity, ImSocket |
| 控制方式 | 人为控制 |

### 3.1.3 接口

|  |  |
| --- | --- |
| 要求 | 1.接口名都以IUpperCamelCase风格编写，**开头需要加I**  2.接口名用名词或者形容词 |
| 正例 | IServer |
| 反例 | Server,server |
| 控制方式 | 人为控制 |

### 3.1.4 Import(插件实现)

|  |  |  |
| --- | --- | --- |
| 要求 | 1. 不要使用通配符 2. 不要换行 3. **分组**，每组使用空格隔开，分组顺序   android包  第三方包  java包  其他 | |
| 正例 | import android.widget.TabHost.OnTabChangeListener;  import com.bangbang.bean.message.MessageType;  import java.io.File; | |
| 反例 | import android.widget.TabHost.OnTabChangeListener;  import com.bangbang.bean.message.MessageType;  import java.io.File; | import android.widget.TabHost.OnTabChangeListener;  import java.io.File;  import com.bangbang.bean.message.MessageType; |
| 控制方式 | 第一点人为控制  第二点，第三点插件控制 | |

### 3.1.5 全局变量

|  |  |
| --- | --- |
| 常量要求 | 1. 全部单词**大写**，单词间用下划线间隔 2. **不是所有final都是常量**，考虑它是否真的感觉像是一个常量 3. 变量长度不得超过20个字符 |
| 变量要求 | 1. 非常量字段名以**lowerCamelCase**风格编写。 2. 静态s开头，非静态m开头 3. 按照**逻辑顺序排列**声明的变量，不要按照时间顺序 4. 变量长度不得超过20个字符 |
| 正例 | final int MAX\_NUMBER = 5;  static int sNumber = 6;  final Set<String> mTreeList = new HashSet<String>(); |
| 反例 | static int maxNumber = 6;  final Set<String> TREE\_LIST= new HashSet<String>();  Int number = 5; |
| 控制方式 | 人为控制 |

### 3.1.6 方法

|  |  |
| --- | --- |
| 命名要求 | 1. 方法名都以**lowerCamelCase**风格编写 2. 方法名通常是动词或动词短语 3. 参数应该**避免用单个字符**命名 |
| 编写要求 | 1. 每个方法不要超过**60行** 2. 返回值为boolean的方法名使用**is(has)前缀** 3. **重载的函数**放在**一起**，中间不要有其他的函数 |
| 正例 | boolean isLeft(int position){  return position > 0?;  } |
| 反例 | boolean getLeft(int p){  return p > 0?;  } |
| 控制方式 | 人为控制 |

### 3.1.7 局部变量

|  |  |
| --- | --- |
| 命名要求 | 1. 局部变量都以**lowerCamelCase**风格编写 2. **避免用单字符**进行命名 3. 即使使用final修饰词，也**不能使用常量的规则**命名 4. 变量长度不得超过20个字符 |
| 编写要求 | 1.**声明时初始化**，几乎所有局部变量声明都应该进行初始化。如果还缺少足够的信息来正确地初始化变量，那就应该推迟声明，直至可以初始化为止。 |
| 正例 | final boolean isKey = false; |
| 反例 | final boolean Key = false;  final String key; |
| 控制方式 | 人为控制 |

### 3.1.8 注释

|  |  |
| --- | --- |
| 注释格式 | 1.**类**要有注释，包括描述、修改记录：作者 修改时间  2.**方法**要有注释，公共方法要有详细的接口输入输出说明  3.**全局变量**要有注释方法中关键逻辑要有注释 |
| javadoc要求 | 1. 每个文件的开头都应该有一句**版权说明** 2. 每个类和自建的**public方法**必须包含Javadoc注释   对于那些无关紧要的类似get和set方法不必撰写Javadoc |
| TODO注释要求 | 1. 对那些临时性的、短期的、够棒但**不完美**的代码，请使用TODO注释 2. 如果TODO注释是“将来要做某事”的格式，则请确保包含一个很明确的**日期**或是一个很明确的**事件** |
| 正例 | /\*\*  \*@param position 当前的位置  \*@return 返回是否当前的位置在屏幕的左边  \*/  public boolean isLeft(int position){  //TODO 2015.9.1前重写判断逻辑  return position > 0?;  } |
| 反例 | public boolean isLeft(int position){  return position > 0?;  } |
| 控制方式 | 人为控制 |

### 3.1.9 语句书写要求

|  |  |  |
| --- | --- | --- |
| 书写点 | 书写要求 | 控制方式 |
| 行长 | 一行代码不要超过**200字** | 插件实现 |
| 换行规则 | 1. 如果在**非赋值**运算符处断开，那么在该符号**前**断开(比如+，它将位于下一行) 2. 如果在**赋值**运算符处断开，通常的做法是在该符号**后**断开(比如=，它与前面的内容留在同一行)。这条规则也适用于foreach语句中的分号   3.方法名或构造**函数名与左括号**留在**同一行**  4.**逗号**(,)与其**前面的内容**留在**同一行**。 | 插件实现能否？ |
| 缩进 | 1. 使用空格缩进，**不要用tab** 2. 自动换行时，第一行后的每一行至少比第一行多缩进4个空格 | 第一点人为控制  第二点插件实现 |
| 空白 | 1. 不同的元素体使用空白隔开 2. 不同的逻辑单元使用空白给开 | 插件目测能实现 |
| 小括号 | 用小括号来**限定组** | 人为控制 |
| 嵌套语句 | 1. Try{}catch{}嵌套语句**不要超过2**行 2. 其他嵌套语句**不要超过3**行 | 人为控制 |
| 异常处理 | 尽量不**要捕获顶级的Exception**，特例：为了阻止错误在用户界面上显示出来，或者保持批量工作的运行时可以考虑捕获顶级的Exception | 人为控制 |

## 资源文件规范（这部分需要记忆）

### Layout中的要求

嵌套不要超过**3层**，IMPORTANT，复杂布局组成部分要有注释

|  |  |  |
| --- | --- | --- |
| **标示符** | **要求** | **示例** |
| **layout** | 1.全部英文小写，单词间以下划线间隔。  2.格式：**业务线\_activity\_功能描述** | House\_activity\_main.xml  house\_activity \_list\_view.xml |
| **Dialog** | 1.全部英文小写，单词间以下划线间隔。  2.格式：**业务线\_dialog\_功能描述** | common\_dialog \_offline.xml |
| **listitem** | 1.全部英文小写，单词间以下划线间隔。  2.格式：**业务线\_listitem\_功能描述** | house\_listitem\_income\_list.xml |
| **Id** | 1.全部小写，单词间以下划线间隔。   |  |  |  |  | | --- | --- | --- | --- | | TextView | tv | EditText | et | | WebView | wv | ImageView | iv | | VideoView | vv | MediaController | mc | | ListView | lv | GridView | gv | | Gallery | gly | Button | btn | | ImageButton | ib | CheckBox | cb | | RadioButton | rb | SeekBar | sb | | ProgressBar | pb | Spinner | spr | | SearchView | sv | AnalogClock | ac | | TimePicker | tp | DatePicker | dp |   2.格式：**layout名\_控件类型\_控件功能** （控件类型参见下表） | house\_main\_btn\_search |

### strings.xml中的要求

|  |  |  |
| --- | --- | --- |
| **标示符** | **要求** | **示例** |
| **Id** | 全部小写，单词间以下划线间隔。Id尽量采取语句英文翻译，或者意义的描述，不要加上前缀后缀。  ok -> 确定  cancel -> 取消  save -> 保存  welcome\_to\_use ->  欢迎使用  use\_at\_first\_time -> 首次使用  在定义前请先到String中搜索之前有没有定义，避免重复定义。 | house\_main\_btn\_search\_label |

### drawable中的要求

|  |  |  |
| --- | --- | --- |
| **标示符** | **要求** | **示例** |
| **文件名** | 全部小写，单词间以下划线间隔。  格式：**类型\_描述\_状态**（只有一种状态则省略）  类型定义：背景(bg)、按钮(btn)、图标(ico)  状态定义：正常(normal)、按下(down)、选中(selected) | bg\_login\_activity.png  btn\_login  ico\_logo  btn\_login\_normal  *btn\_login\_down* |
| **目录使用** | 多张图片放置对应的分辨率缩放目录  **仅一张图片统一放到drawable目录**中 |  |
| **图片后缀名** | 图片后缀名jpg png 需要小写 |  |
| **泛型** | 不要使用 泛型<?,?,?>, ?换成具体的类型 |  |

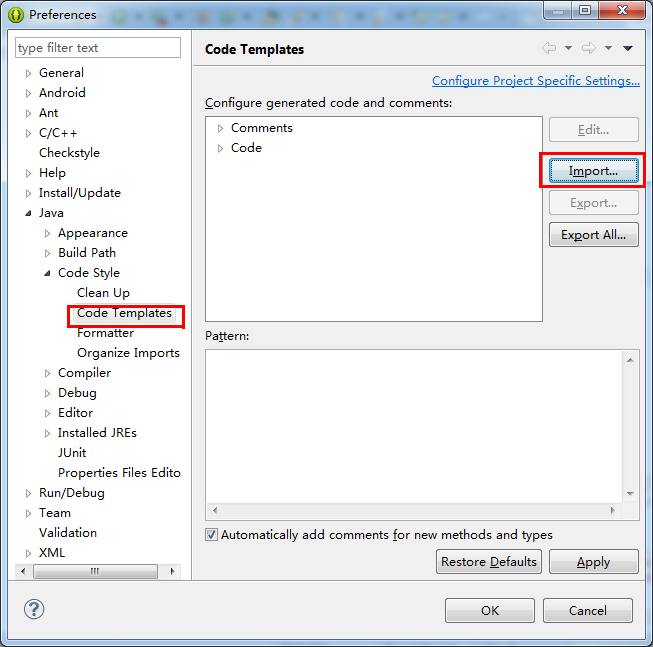
# 辅助工具

## Code Templates

该工具主要用于自动添加注释，包括：文件注释、类注释、方法、变量注释等。代码模板依据Sun标准设置未改动。

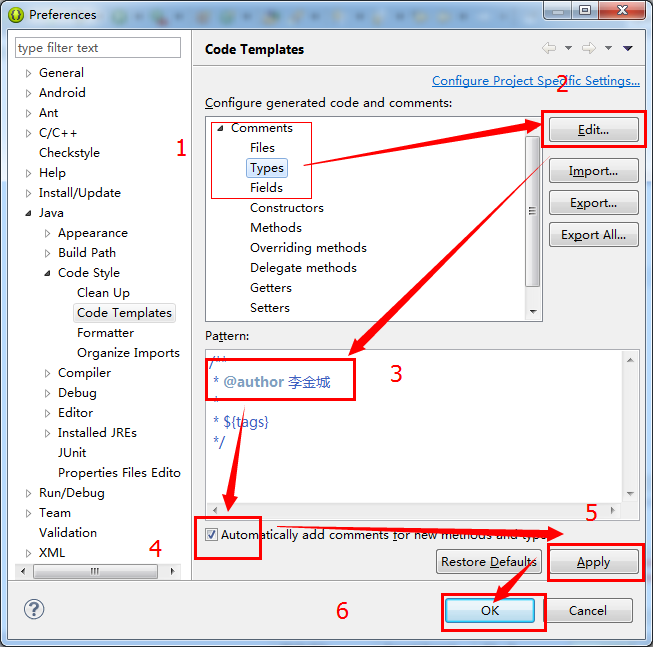
### 导入统一模板

打开窗口：Eclipse🡪Window🡪Perferences🡪Java🡪Code Style🡪Code Templates，如下：



选择Import导入bangbang\_android\_codetemplates.xml

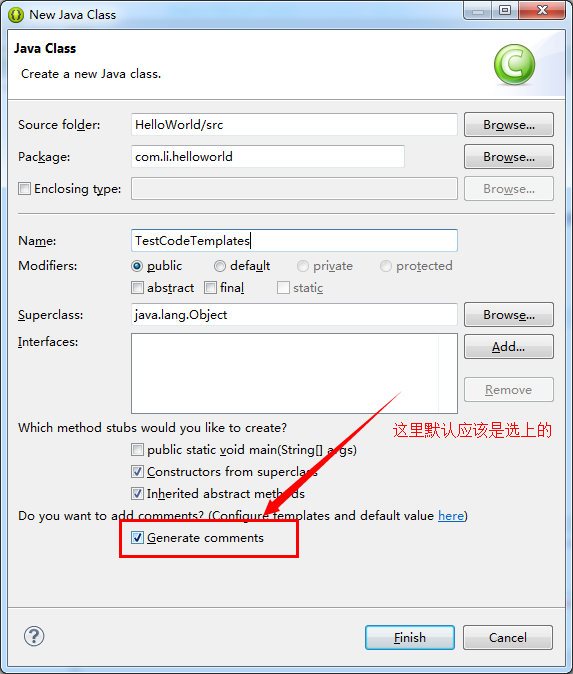
### 修改模板配置



1. 依次打开Comments🡪Types
2. 点击Edit
3. 修改@author 为自己的姓名
4. 选中自动生成注释的复选框
5. 点击Apply和OK按钮

### 验证

创建一个新类，如下图：



生成注释应该是默认选上的（不是默认选上的说明配置不正确），点击finish，创建的类的文件头部注释和类注释如下图表示配置生效。

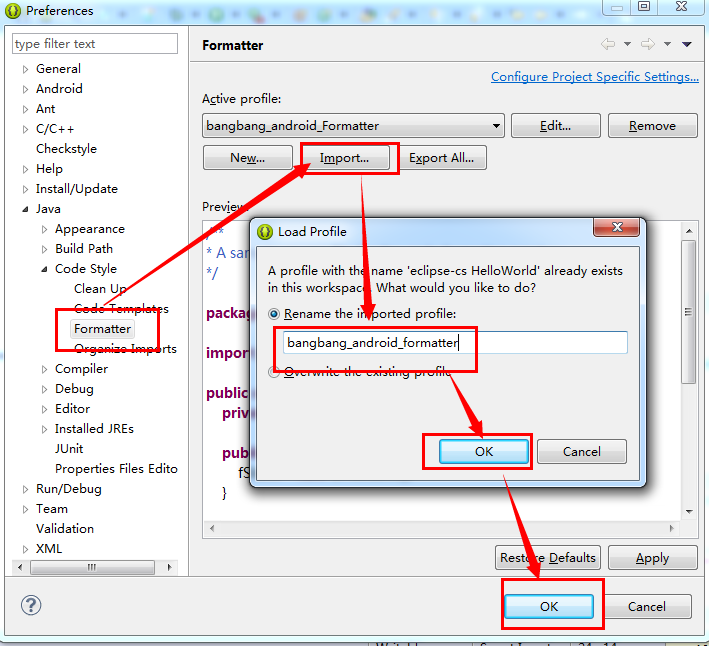


## Formatter

该工具用于格式化我们的代码，保证缩进、换行等风格统一。

### 安装

打开窗口：Eclipse🡪Window🡪Perferences🡪Java🡪Code Style🡪Formatter，如下：



1.点击Import，选择配置文件bangbang\_android\_codeformatter

2.设置格式化名字为bangbang\_android\_formatter

3.点击OK结束

### 使用

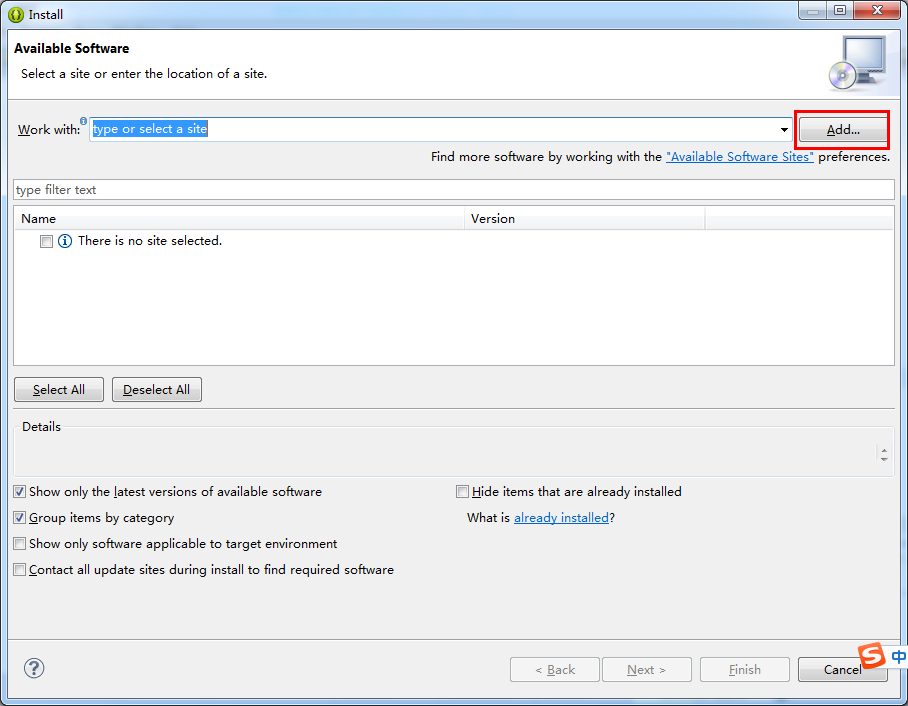
使用时Ctrl+Shift+F即可自动完成格式化。

## Check Style

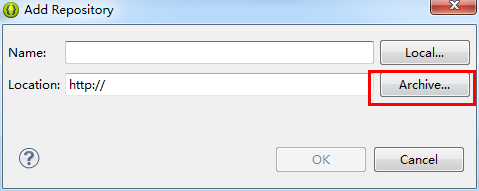
该工具用于自动检查代码格式规范、命名规范等。

### 安装

打开Eclipse🡪Help🡪Install New Software打开如下窗口：



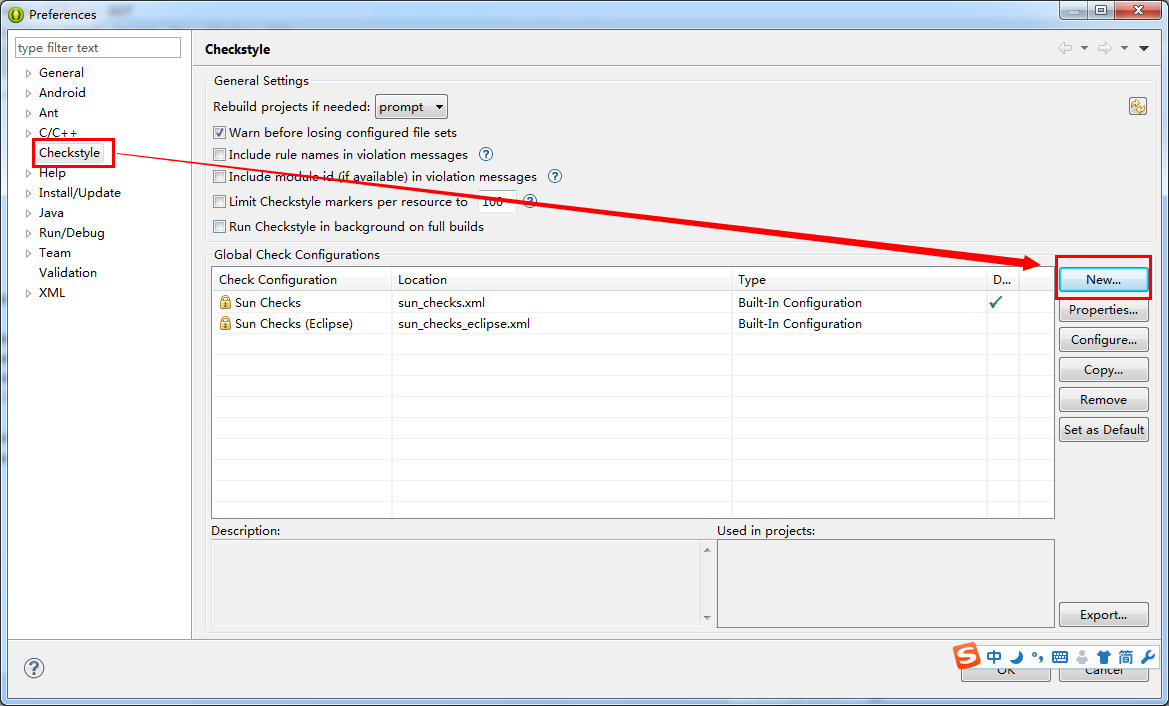
点击ADD按钮，打开如下窗口：



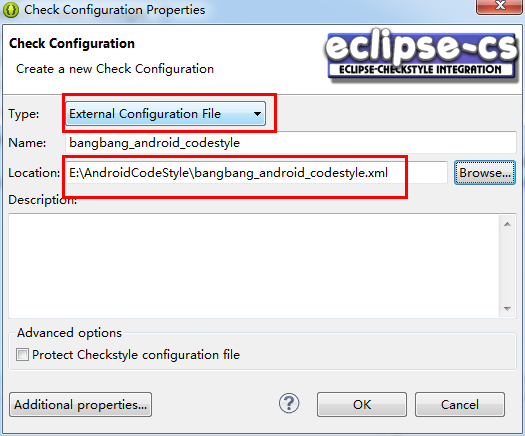
选择Checkstyle\_5.7.0.zip点击OK，然后一路安装下去即可，安装完成会重启，重启后打开Window🡪Preferences 如果出现Check Style则表示插件安装成功。

### 配置

打开窗口Eclipse🡪Window🡪Preferences🡪CheckStyle，如下图：



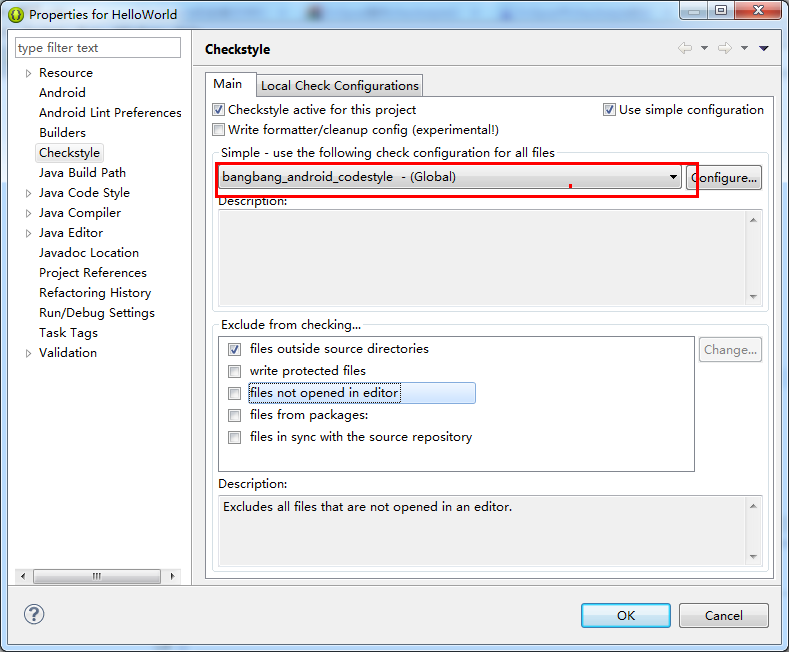
点击New按钮，弹出如下窗口：



选择外部配置文件，设置配置名称并浏览配置文件bangbang\_android\_codestyle.xml位置即完成配置。

### 使用

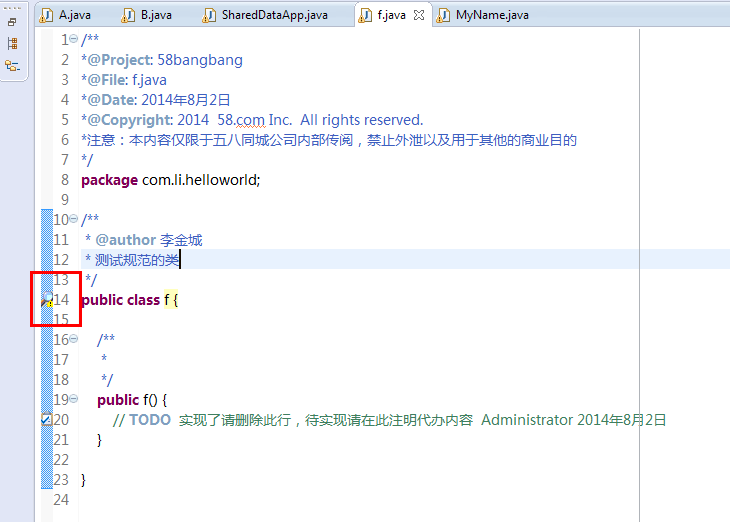
右键项目工程-->Properties—>check Style



选择统一的规范。

工程右键CheckStyle🡪Check Code With CheckStyle进行代码格式的检查。

文件类中出现下图所示的放大镜图标代表代码不符合规范。



## FindBugs

该工具用于检查静态JAVA代码中的潜在问题。

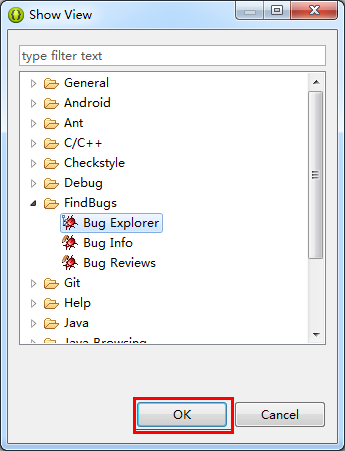
### 安装

1.解压findbugs\_3.0.0.zip

2.复制解压后的文件夹到Eclipse安装目录：D:\Program Files\adt-bundle-windows-x86\_64-20140321\eclipse\plugins 文件夹下。

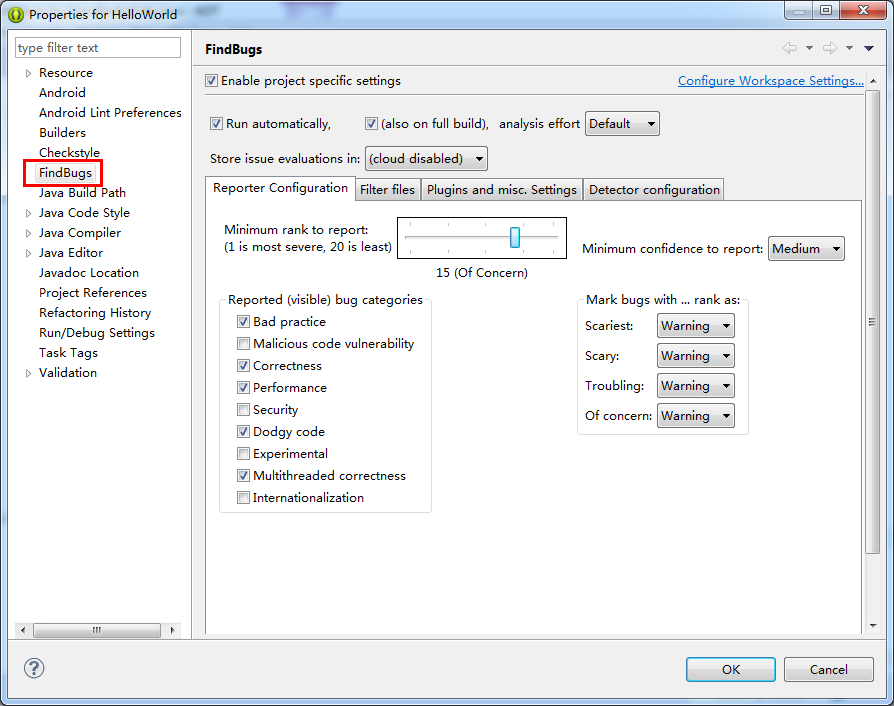
3.重启Eclipse

4.打开Window🡪show View🡪Other中查找FindBugs窗口，如下：



### 配置

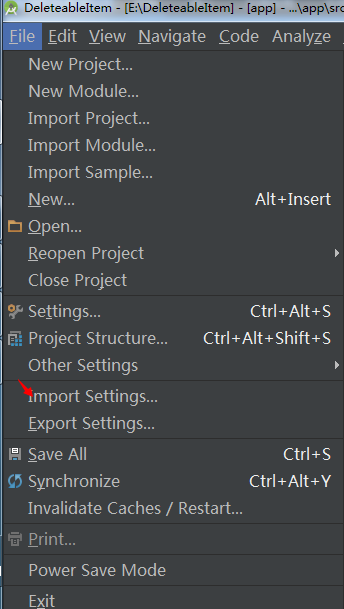
项目右键🡪Properties🡪FindBugs 显示如下窗口：



### FindBugs使用

项目右键🡪FindBugs🡪FindBugs 所有的潜在问题会在BugExplorer中显示

### 4.4.4.导入android studio的setting文件



将settings.jar文件导入，然后重启