参考google官方Java代码规范：

http://www.hawstein.com/posts/google-java-style.html

内容较多，可以慢慢看。

**在本文最后附有代码模板和注释规范.zip中那两个xml文件的导入方式。在elipse中用Ctrl+Shift+F(3个键)来刷新模板，注释则用：/+\*+\*+回车键(4个键)**

这里主要讲述命名，注意命名十分重要，不要大意。

Java代码中名字采用驼峰式：weTogetherGoTo

### 包命名

命名规则：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，edu，gov，mil，net，org。包名的后续部分根据不同机构各自内部的命名规范而不尽相同。这类命名规范可能以特定目录名的组成来区分部门 (department) ，项目(project)，机器(machine)，或注册名(login names)。

*例如： com.welcome.activity*

**规约：包命名必须以com.welcome开始，后面跟有项目名称（或者缩写）,再后面为模块名或层级名称。**

### 类和接口 命名

命名规则：类名是个一名词，采用大小写混合的方式，每个单词的首字母大写。尽量使你的类名简洁而富于描述。使用完整单词，避免缩写词(除非该缩写词被更广泛使用，像 URL，HTML)

接口一般要使用able、ible、er 等后缀

*例如： class Raster; class ImageSprite;*

**规约：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*）。**

*如：class BookMarkAdd 🡪 正确*

*如：class AddBookReadPlanActivity 🡪* ***错误！*** *应为 class BookReadPlanAdd*

### 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 run(); public String getBookName();*

类中常用方法的命名：

1. 类的获取方法（一般具有返回值）一般要求在被访问的字段名前加上get，如

getFirstName()，getLastName()。一般来说，get前缀方法返回的是单个值，find前缀的方法返回的是列表值。

1. 类的设置方法（一般返回类型为void）：被访问字段名的前面加上前缀 set，如

setFirstName(),setLastName().

1. 类的布尔型的判断方法一般要求方法名使用单词 is或has 做前缀，如isPersistent()，isString()。或者使用具有逻辑意义的单词，例如equal 或equals。
2. 类的普通方法一般采用完整的英文描述说明成员方法功能，第一个单词尽可能采用动词，首字母小写，如openFile（），addCount（）。
3. 构造方法应该用递增的方式写。（参数多的写在后面）。
4. toString()方法：一般情况下，每个类都应该定义toString(),其格式为：

### 变量命名

命名规则：第一个单词的首字母小写，其后单词的首字母大写。变量名不应以下划线或美元符号开头，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，即，能够指出其用途。尽量避免单个字符的变量名，除非是一次性的临时变量。临时变量通常被取名为 i，j，k，m 和 n，它们一般用于整型；c，d，e，它们一般用于字符型。

*例如：String bookName;*

**规约：变量命名也必须使用驼峰规则，但是首字母必须小写，变量名尽可能的使用名词或名词词组。同样要求简单易懂，不允许出现无意义的单词。**

*如：String bookName; 🡪 正确*

*如：String bookNameString; 🡪错误！*

### 成员变量命名

同变量命名。

### 常量命名

命名规则：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 = 4;

例如：static final int MAX\_WIDTH = 999;

例如：static final int GET\_THE\_CPU = 1;

### 异常命名

自定义异常的命名**必须以Exception为结尾**。已明确标示为一个异常。

### layout 命名

**规约：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：layout\_knowledge\_gained\_main.xml🡪正确*

*如：list\_book.xml🡪错误！*

### id 命名

**规约：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。**

*如：某TextView @+id/textbookname 🡪错误 !应为 @+id/tv\_book\_name\_show*

*如：某EditText @+id/textbookname 🡪错误 !应为 @+id/et\_book\_name\_edit*

### 资源命名

**规约：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**

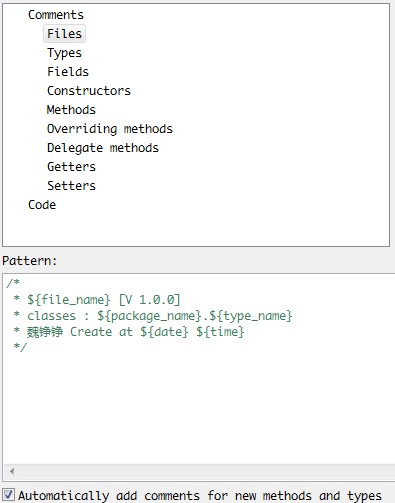
*如：menu\_icon\_navigate.png 🡪 正确*

*如：某分割线：line.png 或 separator.png 🡪正确*

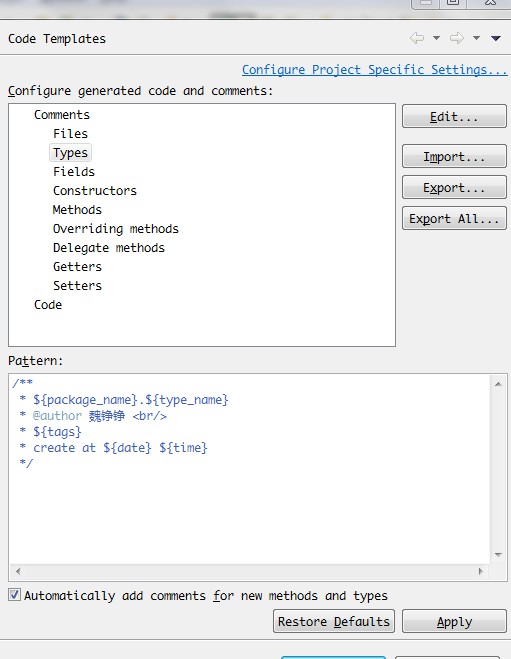
### 注释模板

在eclipse 的preferences 中，选择 java 🡪 code style 🡪code Template

1. 添加文件创建日志模板



2.设置类注释模板



### 导入方法

在eclipse 的preferences 中，选择 java 🡪 code style 🡪code Template 中选择 Import，选择附件中的文件。



**但是注意修改作者为自己：${user} (指向计算机名称)或者你的名字**

### 格式化模板

在eclipse 的preferences 中，选择 java 🡪 code style 🡪formatter 中选择 Import，选择附件中的文件。