

项目编码规范

命名规范:

- 局部/临时变量: **variableName**
-
- 公有函数: **FunctionName()**
-
- 私有/保护函数: **_PrivateFunction()**
-
- 常量/枚举: **MACRO_NAME**
-
- 变量成员: **mMemberName**
- 类名: **ClassName**
- 接口命名
- 接口名使用 **I** 打头加名词的形式每个单首字母大写, 其他字母小写 **ITest**
- 函数、类、接口、变量的命名都应该尽可能简单易懂, 让人一看就知道那个是干什么的。
- 尽量不要缩写或者自己 **YY**。
- 例如: 计数器写成 **count** 而不是 **c**。(a\b\c 这种单个字母的是最不应该出现的, 除了大家都懂的在循环时用的 **i**)

代码风格:

过长的代码(超出屏幕)要分成 2 行。

代码块(类、函数、判断、循环等等):

```
void FunctionName()  
{  
    cpp way;  
}
```

空实现的函数:

```
void FunctionName() {}
```

其他:

- **Commit** 你的代码之前先确定能够编译通过(最好自己先做一次测试)。
- 充分注释。
- 对每个变量使用最合适的数据类型。(例如要是你的一个变量数值只从 -10~10 的话, 就不要用 **int**, 而是用 **byte**, 减少内存浪费)。
- **Using\import** 的顺序: 首先是我们自己写的类, 隔一行, 第三方库里的类, 隔一行, 系统本身的类。

JAVA 注释(C#也适用)

1. 文件注释

文件注释放在文件头部, 主要包括文件的功能说明, 编写人和修改人以及编码时间和修改时间, 版权声明, 版本等信息。

格式:

```
/**
```

```
* @file 文件名
* 文件说明
* @author
* - 时间 作者 1 贡献 1
* - 时间 作者 2 贡献 2
* 版权信息
*/
```

例子:

```
/**
 * @file DemoComment.java
 * 文件说明
 * @author
 * - 2011-01-02 Lee 创建初始版本
 * - 2011-01-05 Leeforall 添加 getAppName() 方法
 *
 * @License: Apache License 2.0 http://www.apache.org/licenses/LICENSE-2.0
 */
```

2. 类注释

类注释放置在类声明前，主要介绍类的功能及相关说明。

```
/**
 * 类功能说明
 */
public class Demo
{
}
```

3. 方法注释

方法注释放置在其声明前，主要介绍函数的功能、参数、返回值、异常、使用说明

格式:

```
/**
 * 方法功能说明
 * @param 参数名称 1 简要说明;
 * @param 参数名称 2 简要说明。
 * @return 返回值 返回值说明
 * @throws 异常说明
 * - 异常 1 异常 1 说明;
 * - 异常 2 异常 2 说明。
 * @example 输出或返回值示例(可选)
 * - 时间 作者 1 贡献 1
 * - 时间 作者 2 贡献 2
 */
```

示例:

```
/**
 * 获得好友列表
 *
 * @param usrId
 * 用户 ID
 * @return xml 格式的好友信息列表
 */
```

```

* @example <root> <contacter userId="Lee" userName="Lee"
* userEmail="leeforall@qq.com" userBrief="Lee"
* onlineFlag="1"/>
*
* <contacter userId="bao" userName="Lee"
* userEmail="leeforall@qq.com" userBrief="Lee"
* onlineFlag="0"/> </root>;
* @author -2011-01-10 Lee 添加
*/
public String getFriends(String usrId)
{
}

```

4. 变量注释

变量统一使用块注释

格式:

```

/**
 * 变量说明
 */
public String demo;

```

5 语句块注释

应使用单行注释，即//

一目了然的语句不应该注释；

每个分支、每个功能段均应注释

例子:

```

if (count > 10)
{ //大于 10 处理
    count = 0;
}
else
{ //小于等于 10 处理
    count++;
}
// 计算输出长度
var length:int = count * 10;

```

历史更新记录表

版本号	更新日期	简述	修订人
Version 0.1.0	2012-3-3	确定初版编码规范	李源东
Version 1.0.0	2012-3-4	确定最终版编码规范	李灏