代码编写规范

——基于Flutter的移动端跨平台记账APP



项目名称： 账单记录APP

专业班级： 软件工程1802

小组成员：徐任、牟灵成、莫丁阳

指导教师： 杨枨老师

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 修订日期 | 版本/状态 | 修订人 | 备注 |
| 01 | 2020.12.15 | 0.1 | 牟灵成 |  |
| 02 | 2020.12.20 | 0.2 | 徐任 |  |
| 03 | 2020.12.28 | 0.3 | 徐任 |  |

# 目录

[代码编写规范 1](#_Toc60098737)

[目录 2](#_Toc60098738)

[1. 目的 3](#_Toc60098739)

[2. 概述 3](#_Toc60098740)

[3. 代码规范 3](#_Toc60098741)

[3.1 有关命名 3](#_Toc60098742)

[3.1.1项目命名 3](#_Toc60098743)

[3.1.2包命名 3](#_Toc60098744)

[3.1.3 类命名 3](#_Toc60098745)

[3.1.4 方法命名 3](#_Toc60098746)

[3.1.5 变量命名 4](#_Toc60098747)

[3.1.6 常量命名 4](#_Toc60098748)

[3.1.7 资源文件命名 4](#_Toc60098749)

[3.1.8 其他命名 4](#_Toc60098750)

[3.2 有关注释 4](#_Toc60098751)

[3.2.1 程序文件头注释 4](#_Toc60098752)

[3.2.2 方法头注释 5](#_Toc60098753)

[3.2.3 关键点注释 5](#_Toc60098754)

[4. 格式规范 5](#_Toc60098755)

[4.1 缩进 5](#_Toc60098756)

[4.2 换行 6](#_Toc60098757)

[4.3 对齐 6](#_Toc60098758)

[5. 写在后面 6](#_Toc60098759)

[6. 补充 6](#_Toc60098760)

# 1. 目的

         为了使软件开发过程顺畅，保证软件质量，G14小组编写这份开发规范文档。

# 2. 概述

* 项目以功能模块来划分分工；
* 数据库不能随意修改，本地数据库须统一；
* 时刻考虑代码的可复用性；

# 3. 代码规范

## 3.1 有关命名

总体原则上，变量名与方法命名时应该遵循良好的命名习惯，尽量使得命名能够体现变量或者方法本身的含义。

名称只能由数字、字母、下划线、$符组成，不能以数字开头

尽量不要使用拼音 或者 英语掺杂拼音

### 3.1.1项目命名

项目名全部采用小写方式命名

### 3.1.2包命名

包名全部采用小写方式命名

### 3.1.3 类命名

类名首字母大写，若类名由多个单词构成，每个单词首字母大写，即驼峰命名法

### 3.1.4 方法命名

方法名首字母小写，若其由多个单词构成，除第一个单词之外，其他单词首字母大写,即Camel命名法

### 3.1.5 变量命名

所有的变量名都小于32字符

变量名首字母小写，若其由多个单词构成，除第一个单词之外，其他单词首字母大写,即Camel命名法

### 3.1.6 常量命名

常量名全部采用大写方式命名

### 3.1.7 资源文件命名

参考G14版本命名规则

### 3.1.8 其他命名

## 3.2 有关注释

       团队成员都应该形成良好的写注释的习惯，方便以后阅读。

### 3.2.1 程序文件头注释

应该包含如下：

\* 文件描述 （Description）：描述此类的作用；

\* 作者 （Author）：创建者或者修改者名；

\* 版本 （Version）：创建或者修复时的编号，需要自行在bug管理系统中创建bug 号，使用bug号进行命名（若无bug管理工具的临时办法：如无bug号，从1开始，修改时依次增加）

\* 日期（Date）：创建或者修改时的日期，使用“-”进行年月日分割；

\* 记录（Record）：创建或者修改的工作内容描述；

提前设置好文件的模板Template， 模板以下：

/\*

\* Description:

\* Author Version Date Record

\* KevinLee 1 2017-11-7 版本创建

\*/

### 3.2.2 方法头注释

       一般在写完一个方法后使用快捷键生成一个块注释，IDE会自动帮我们写入一些信息。   
应该包含如下信息：   
       \* 方法描述 Description:   
       \* 参数信息 @param   
       \* 返回信息 @return   
       \* 异常信息 @Exception   
       如以下模板：

/\*\*

 \* Description: 返回一个“Hello”字符串

 \* @param str 一个字符串

 \* @return 返回一个字符串

 \* @throws Exception  抛出一个异常

 \*/

public  String  sayHello(String  str) throws  Exception{

    str="Hello";

  return  str;

}

注意：方法里面不要使用块注释

### 3.2.3 关键点注释

应该包含如下信息：  
\* 一些程序关键的地方；  
\* 一些程序不易读的地方；  
\* 编写代码过程中遇到问题的地方；  
\* 需要提示读者的地方；

# 4. 格式规范

## 4.1 缩进

代码缩进为两个制表符

## 4.2 换行

{}花括号应该另起一行，左花括号与方法名、类名在同一行。(除了数组初始化时的花括号)；

if、while等语句，假如体内只有一句代码也不要省略{}，为了方便以后的增删；

字符串过长考虑拆分成多行；

{}括号等应该对齐；

类和方法的块注释必须紧贴类和方法；

单独起行的//注释必须对齐被注释语句；

## 4.3 对齐

{}之内的代码要向内缩进一个 Tab，{与上一句代码同行，}单独成行，地位不同的继续缩进

# 5. 写在后面

       希望各位成员遵守这份开发规范文档，养成良好的开发习惯；

# 6. 补充

      暂无