|  |
| --- |
|  |
| iOS编码规范 |
| 四合科技 |
|  |
| **谢怀清** |
| **2016/4/6 Wednesday** |

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修正人** | **说明** | **日期** |
| V1.0 | 谢怀清 | 初始版本 | 2016-04-06 |

目录

[**一、目的** 2](#_Toc447723389)

[**二、适用范围** 2](#_Toc447723390)

[**三、编码规范** 2](#_Toc447723391)

[3.1 文件组织结构 2](#_Toc447723392)

[3.1.1源文件组织 2](#_Toc447723393)

[3.1.2图片资源文件组织 2](#_Toc447723394)

[3.2命名规范 2](#_Toc447723395)

[3.2.1类命名 2](#_Toc447723396)

[3.2.2方法命名 2](#_Toc447723397)

[3.2.3变量命名 2](#_Toc447723398)

[3.2.4常量命名 3](#_Toc447723399)

[3.3编码排版规范 3](#_Toc447723400)

[3.3.1类结构排版 3](#_Toc447723401)

[3.3.2方法结构排版 4](#_Toc447723402)

**一、目的**

统一规范Xcode编辑环境下Objective-C的编码风格。

**二、适用范围**

适用于本公司所有使用Objective-C开发的项目。

**三、编码规范**

3.1 文件组织结构

3.1.1源文件组织

项目工程文件结构与物理结构保持一致，源文件组织要遵循一下两大原则：

1. 基于MVC设计模式原则，各部分对应自己的文件目录；
2. 基于功能模块划分原则，各模块对应自己的文件目录；

3.1.2图片资源文件组织

资源文件使用Images.xcassets管理，基于功能模块划分对应的文件目录。

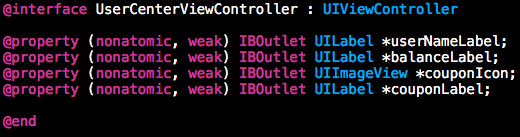
3.2命名规范

所有源文件命名遵循驼峰命名法，图片资源文件命名使用全小写字母，使用”\_”分割命名。

3.2.1类命名

类名应该使用首字母大写的驼峰命名法，在应用级别的代码里，不要使用带前缀的类名。

例如：



Category命名应使用类名+扩展。

例如：



3.2.2方法命名

方法应该使用首字母小写的驼峰命名法，执行性的方法应该以动词开头。一个规范的方法读起来应该像一句完整的话，能够完整的表达出函数的作用。

例如：



3.2.3变量命名

变量名应该使用首字母小写的驼峰命名法，使用有意义的、可以推测其用途的单词。

例如：

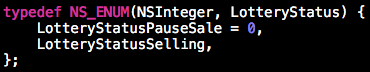


3.2.4常量命名

预定义的常量使用小写k开头的驼峰命名法，例如：



枚举型常量使用大写字母开头的驼峰命名法，例如：



局部常量使用大写字母开头的驼峰命名法，例如：



3.3编码排版规范

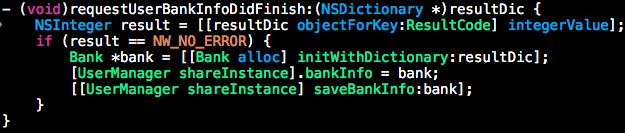
3.3.1类结构排版

类的生命周期方法放在类的最上面，公共方法放在这之后，私有方法放在类的最下方。方法之间使用空行分隔，相同或相近功能的方法采用”#pragma mark –“ 来分割标记，以便查看。例如：



3.3.2方法结构排版

方法标识”-/+”后空格，”{”不要单独占用一行与代码直接用空格分隔。不同逻辑块直接可以使用空行分隔。例如：



3.3.3其他

删除多余的空行，删除多余的空格，删除多余的注释，删除多余的方法，删除一切多余的东西。