

天天爱K歌项目编程规范

本项目共使用了4种编程语言或框架：用户手机端app使用**Java**编写，服务器端使用**Python**编写，管理员歌曲上传系统由**Vue**框架搭建，歌唱打分算法使用 **C++** 实现。本文件将分别对这4种编程语言或框架进行编程规范的描述。

0. 目录

- 1. Java
 - 1.1. 命名
 - 1.2. 源文件结构
 - 1.3. 注解
 - 1.4. 括号
 - 1.5. 缩进
 - 1.6. 空行
 - 1.7. 空格
 - 1.8. 语句
- 2. Python
 - 2.1. 命名
 - 2.2. 括号
 - 2.3. 缩进
 - 2.4. 空行
 - 2.5. 空格
 - 2.6. 语句
- 3. Vue
 - 3.1. 命名
 - 3.2. 缩进
 - 3.3. 组件规范
- 4. C++
 - 4.1. 命名
 - 4.2. 源文件命名
 - 4.3. 注解
 - 4.4. 格式
 - 4.5. 行长度

1. Java

本项目使用**Java编程语言**进行用户手机端app的搭建，其编程规范如下：

1.1. 命名

- 包名全部小写，连续的单词只是简单地连接起来，不使用下划线。
- 类名和自定义类型名以 UpperCamelCase 风格编写。
- 方法名以 lowerCamelCase 风格编写。
- 常量命名模式为 CONSTANT_CASE，全部字母大写，用下划线分隔单词。
- 参数名以 lowerCamelCase 风格编写。
- 非常量字段名、参数名和局部变量名以 lowerCamelCase 风格编写。避免用单字符进行命名，除了临时变量和循环变量。

1.2. 源文件结构

一个源文件包含(按顺序地)：

- 许可证或版权信息(如有需要)
- package语句
- import语句
- 一个顶级类(只有一个)。

以上每个部分之间用一个空行隔开。这其中：

- package 语句不换行。
- import 语句不换行，且不使用通配符 *。
- import 语句可分为以下几组，按照这个顺序，每组由一个空行分隔：
 - 所有的静态导入独立成组
 - com.google imports(仅当这个源文件是在 com.google 包下)
 - 第三方的包。每个顶级包为一组，字典序。例如： android， com， junit， org， sun
 - java imports
 - javax imports组内不空行，按字典序排列。
- 顶级类名与文件名相同。

1.3. 注解

- 在可能时使用 @Override 注解

- 每个注解独占一行

1.4. 括号

- 即使块中只有一条语句，也应该使用大括号。
- 大括号遵循Kernighan and Ritchie风格：
 - 左大括号前不换行
 - 左大括号后换行
 - 右大括号前换行
 - 如果右大括号是一个语句、函数体或类的终止，则右大括号后换行；否则不换行。

1.5. 缩进

- 使用4个空格缩进代码。

1.6. 空行

- 对多行空格不做限制。
- 以下情况需要使用空行：
 - 类内连续的成员之间：字段，构造函数，方法，嵌套类，静态初始化块，实例初始化块。
例外：两个连续字段之间的空行是可选的，用于字段的空行主要用来对字段进行逻辑分组。
 - 在函数体内，语句的逻辑分组间使用空行。

1.7. 空格

- 空格应该出现在代码中的以下位置：
 - 分隔任何保留字与紧随其后的左括号
 - 分隔任何保留字与其前面的右大括号
 - 在任何左大括号前
 - 在任何二元或三元运算符的两侧
 - 在 , : ;) 及右括号后
 - 类型和变量之间。

1.8. 语句

- 每个语句独占一行。
- 数组初始化可写成块状结构。
- 一个语句只声明一个变量。
- `switch` 语句中必须包含 `default` 。
- 类和成员的modifiers如果存在，则按Java语言规范中推荐的顺序出现。

public protected private abstract static final transient volatile synchronized native strict

2. Python

本项目使用**Python编程语言**进行服务器端的搭建，其编程规范如下：

2.1. 命名

- 使用下划线命名法进行变量的命名。
- 避免双下划线开头并结尾的变量名称。

2.2. 括号

- 除非用于实现行连接, 否则不在返回语句或条件语句中使用括号。

2.3. 缩进

- 使用4个空格缩进代码。

2.4. 空行

- 不同方法之间空两行。
- 每个方法中逻辑连续性较低的代码块之间空一行。

2.5. 空格

- 在逗号, 分号, 冒号后面加一个空格。
- 参数列表, 索引或切片的左括号前不加空格。
- 在二元操作符两边各加一个空格。

2.6. 语句

- 每个语句独占一行。

3. Vue

本项目使用**Vue框架**进行管理员歌曲上传系统的搭建，其编程规范如下：

3.1. 命名

- 单文件组件的文件名使用Pascal命名法。
- 组件名称为多个单词。
- 变量名称使用驼峰命名法。
- `<template>` 中的标签使用中划线进行单词的连接。

3.2. 缩进

- 使用2个空格缩进代码。

3.3. 组件规范

- 组件的 `data` 必须是返回一个对象的函数。
- 组件的 `prop` 的定义需要指明其类型。
- 使用 `v-for` 时为其指定键值。
- 为组件的样式设置作用域。
- 将每个组件分为单独的文件。

4. C++

本项目使用 **C++** 实现歌唱打分算法，采用google C++ 编码规范。其编程规范（节选）如下：

4.1. 命名

- 函数命名, 变量命名, 文件命名要有描述性; 少用缩写。
- 类型名称的每个单词首字母均大写, 不包含下划线: `MyExcitingClass` , `MyExcitingEnum` 。
- 变量 (包括函数参数) 和数据成员名一律小写, 单词之间用下划线连接. 类的成员变量以下划线结尾, 但结构体的就不用, 如: `a_local_variable` , `a_struct_data_member` , `a_class_data_member_` 。
- 声明为 `constexpr` 或 `const` 的变量, 或在程序运行期间其值始终保持不变的, 命名时以 “k” 开头, 大小写混合。
- 常规函数使用大小写混合, 取值和设值函数则要求与变量名匹配: `MyExcitingFunction()` , `MyExcitingMethod()` , `my_exciting_member_variable()` , `set_my_exciting_member_variable()` 。
- 枚举的命名应当和 **常量** 或 **宏** 一致: `kEnumName` 或是 `ENUM_NAME` 。

4.2. 源文件命名

文件名要全部小写, 可以包含下划线 (`_`) 或连字符 (`-`) 。

4.3. 注解

- 每个注解独占一行。

- 每个类的定义都要附带一份注释, 描述类的功能和用法, 除非它的功能相当明显。
- 对于代码中巧妙的, 晦涩的, 有趣的, 重要的地方加以注释。

4.4. 格式

- 返回类型和函数名在同一行, 参数也尽量放在同一行, 如果放不下就对形参分行, 分行方式与 [函数调用](#) 一致。
- 使用4个空格缩进代码。
- 要么一行写完函数调用, 要么在圆括号里对参数分行, 要么参数另起一行且缩进四格。如果没有其它顾虑的话, 尽可能精简行数, 比如把多个参数适当地放在同一行里。
- 倾向于不在圆括号内使用空格。关键字 `if` 和 `else` 另起一行。
- `switch` 语句可以使用大括号分段, 以表明 cases 之间不是连在一起的。在单语句循环里, 括号可用可不用。空循环体应使用 `{}` 或 `continue`。
- 水平留白的使用根据在代码中的位置决定。永远不要在行尾添加没意义的留白。
- 垂直留白越少越好。

4.5. 行长度

每一行代码字符数不超过 80。