Java编程规范

**天津爱迪尔软件开发有限公司**

**修正记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **作者** | **描述** |
| **1.0** | **2014-03-26** | **Jerry** | **初稿** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. 概述
   1. **背景**

程序编码是一个极具创造性的工作,虽然要奇思妙想，但是也要遵循一定的规则和限制.编码风格的重要性不言而喻

开发工程师在开发过程中必须遵循本规范,规范是代码编写及验收等管理环节中必须执行的标准.

* 1. **读者对象**

本手册阅读对象为Java开发工程师.

1. **编程规范**
   1. 公共约定

命名是指对源文件中文件名，包，方法，数据库表，数据库字段等的命名.

所有的命名用英文来编写

1.文件名（类名）使用驼峰命名,如果LoginAction 单词首字母大写.业务逻辑层的接口以BS、Biz结尾实现类则是以BSImpl、BizImpl结尾。

2.包名使用全部小写,域名逆置如域名为www.idea.com 则使用com.idea开头.项目名紧跟其后,然后包名再分类为先功能后MVC ，例如 com.idea.itms.portal.action portal 为功能,action为项目中的MVC分层.实现类的包名以impl结尾.多加一个common包放一些该功能的一些工具包。

3MVC分层一般分为action ,service,data(还可对data包进行分类),dao(可封装)

4实现接口类一律使用impl结尾,如果接口为StaffBS则实现类为StaffBSImpl,包名应该为com.portal.service.impl

5方法名使用驼峰命名，首字母小写，例如queryVOById 简称或者大家公认的简写可以大写使用.

6 简称可以用大家公认的，比如mgr -manager 等

7 数据库表命名,全部为小写，[模块名]\_[功能名] 模块名简称为1-3个字母,如bm\_staff

视图一律以v\_开头，存储过程以p\_ 触发器以t\_开头索引以i\_开头

8 数据库字段名命名，全部为小写，[功能名]\_[字段] staff\_id

9 变量名名，一个单词的全部为小写，2个单词以上一律驼峰命名，首字母小写，例如staffId,staffName这样的命名。若为常量则定义为static final 类型，全部大写字母，若大于2个单词使用下划线分开例如COMMON\_STATE

* 1. 其他命名

组件命名: 匈牙利命名 如lblName 前缀+ID

对象类型 前缀

Canvas cvs

CheckBox chk

Image img

List(组件) lst

Choice chc

Dialog dlg

Event evt

Frame frm

Menu menu

Panel pnl

TextArea txa

TextField txf

* 1. 集合命名

集合Set，List 使用集合名结尾，例如 staffList

数组命名 则为复数 如staffs

* 1. JSP命名

jsp命名全部小写，使用\_分开 例如 staff\_list.jsp 即可能的去说明JSP的功能。

jsp文件夹存放也要按照src中Java的功能分类即可例如portal包放所有该功能的JSP

* 1. Servlet命名

所有的servlet命名使用\*Action这样的，比如LoginAction ，servlet的URL设置为，/功能名/\*Action.html

* 1. 对齐

建议4个空格为一个单位，不要一味的使用Tab进行排版，电脑上Tab键设置空格数量不一定.

花括号{}上下对齐

1. 注释

所有的注释是英文注释,内容只要涵盖意思即可.此项为硬性要求.

* 1. 文件头注释

/\*\*

\* Filename: [文件名]

\* Description: This is a java file,porcess staff action.

\* Copyright: Copyright © 2012 - 2020 Idea Innovative Technology Ltd.All Rights Reserved.

\* Author: [作者]

\* ModifyBy: [修改人]

\* ModifyDate：[修改时间]

\* ModifyContext: [修改内容]

\*/

* 1. 类与接口注释

/\*\*

\* <一句话功能简述>

\* <功能描述>

\* @author [作者]

\* @version[版本号,YYYY-MM-DD]

\* @see[相关类,相关方法]

\* @since [产品、模块版本]

\* @deprecated [该注释表示该方法已经废弃或者不建议使用]

\*/

* 1. 方法注释

/\*\*

\* <一句话功能简述>

\* <功能描述>

\* @param [参数1] [参数1说明]

\* @param [参数1] [参数1说明]

\* @return [返回类型说明]

\* @exception/throws [异常类型] [异常说明]

\* @see[相关类,相关方法]

\* @deprecated [该注释表示该方法已经废弃或者不建议使用]

\*/

* 1. 其他注释

字段只需要行内注释//即可，若期中的功能需要说明在代码内也是要行内注释

//如果flag为真那么。。。。

if（flag）｛

｝

  对get,set这样的方法不用进行注释.

数据库的表名字段名也要进行注释，只需要说明其意义即可.

1. 开发注意事项
   1. 单例对象使用注意

使用饿汉单例，类对象直接实例化，getinstance时候直接返回类对象  空间转换时间  节省了运行时间。

public class Singleton{

    private static class SingletonHolder{

    private static Singleton instance = new Singleton();

    }

    private Singleton(){}

    public static Singleton getInstance(){

        return SingletonHolder.instance;

    }

}

在有必要时候 使用枚举代替单例、

* 1. 泛型

列表优于数组，数组扩展性不好。使用List ArrayList 性能更优

Map中弱使用有序的Map则使用LinkendMap

接口优于抽象类

* 1. 通用程序设计

foreach循环优先于传统的for循环

避免使用Fload Double