**客户名字和logo**

[项目名\_文档名]

[项目ID]

Zhi Luo

日期：2015-01-12

文档版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修订日期 | 修订人 | 审核人 | 变更内容 |
| 0.1 | 2015-01-08 | LUO ZHI |  | 初始化 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 文档介绍 1](#_Toc408827924)

[1.1 目的 1](#_Toc408827925)

[1.2 范围 1](#_Toc408827926)

[1.3 缩写词列表 1](#_Toc408827927)

[1.4 参考内容 1](#_Toc408827928)

[2 命名规范 2](#_Toc408827929)

[2.1 驼峰命名 2](#_Toc408827930)

[2.2 源文件命名及编码 2](#_Toc408827931)

[2.3 包命名规范 2](#_Toc408827932)

[2.4 类命名规范 2](#_Toc408827933)

[2.5 方法命名规范 2](#_Toc408827934)

[2.6 常量命名规范 2](#_Toc408827935)

[2.7 变量命名规范 3](#_Toc408827936)

[2.8 参数命名规范 3](#_Toc408827937)

[3 源文件结构 3](#_Toc408827938)

[3.1 项目目录结构 3](#_Toc408827939)

[3.2 包引入规范 3](#_Toc408827940)

[3.3 类声明规范 3](#_Toc408827941)

[3.4 注释规范 3](#_Toc408827942)

[3.4.1 源文件注释 3](#_Toc408827943)

[3.4.2 类注释 4](#_Toc408827944)

[3.4.3 变更注释 4](#_Toc408827945)

[3.4.4 方法注释 4](#_Toc408827946)

[3.4.5 代码内注释 4](#_Toc408827947)

[3.4.6 Javadoc 4](#_Toc408827948)

[4 代码排版 4](#_Toc408827949)

[4.1 括号 4](#_Toc408827950)

[4.2 缩进 5](#_Toc408827951)

[4.3 每行字符限制 5](#_Toc408827952)

[4.4 单行语句 5](#_Toc408827953)

[4.5 代码行限制 5](#_Toc408827954)

[5 代码书写规范 5](#_Toc408827955)

[5.1 简单语句 5](#_Toc408827956)

[5.2 复杂语句 5](#_Toc408827957)

[5.3 返回语句 5](#_Toc408827958)

[5.4 条件语句 5](#_Toc408827959)

[5.5 分支语句 5](#_Toc408827960)

[5.6 循环语句 5](#_Toc408827961)

[5.7 异常控制 5](#_Toc408827962)

[5.8 访问控制 5](#_Toc408827963)

[6 日志规范 5](#_Toc408827964)

[6.1 禁止使用System.out.println 5](#_Toc408827965)

[6.2 Log4j配置 5](#_Toc408827966)

[6.3 日志等级 5](#_Toc408827967)

[6.4 日志分类 5](#_Toc408827968)

[6.5 日志格式 5](#_Toc408827969)

[6.6 日志定义与继承 5](#_Toc408827970)

[7 编程惯例 6](#_Toc408827971)

[7.1 常量 6](#_Toc408827972)

[7.2 异常按层次抛出 6](#_Toc408827973)

[7.3 国际化 6](#_Toc408827974)

[7.4 字符串生成 6](#_Toc408827975)

[7.5 Model类建议实现接口 6](#_Toc408827976)

[7.6 面向接口编程 6](#_Toc408827977)

[7.7 避免使用自带同步机制的集合类 6](#_Toc408827978)

[7.8 变量初始化 6](#_Toc408827979)

# 

# 文档介绍

## 目的

本文档描述用于eOrder订餐系统实现代码的编码规范。

## 范围

本文档描述内容包括地eOrder订餐系统实现代码的代码命名，排版，注释，语句以及日志的编写规范。本文档的不包括任何eOrder订餐系统的系统框架，设计和实现的任何部分。本文仅供eOrder订餐系统开发成员使用。

## 缩写词列表

n/a

## 参考内容

|  |  |
| --- | --- |
| 文献名称 | 文献资源 |
| 谷歌编码规范 | http://google-styleguide.googlecode.com/svn/trunk/javaguide.html |
| SUN JAVA编码规范 | http://www.oracle.com/technetwork/articles/javase/codeconvtoc-136057.html |
|  |  |

# 命名规范

## 驼峰命名

eOrder订餐系统Java实现代码均使用驼峰命名规则。驼峰命名规则用户类，函数名，变量名定义时，如果使用多个英文单词连接，将单词首字母大写，形势骆驼的驼峰而得名。例如：eOrderService。

在形式上驼峰命名又分为小驼峰和大驼峰，区别在于第一个单词的首字母是否大写。如果大写，则是大驼峰命名；反之，则是小驼峰命名。

本文约定eOrder订餐系统中，源文件，类，接口定义均使用大驼峰命名；而方法，变量使用小驼峰。

本文唯一例子是常量的命名采用全大写希腊式命名。希腊式命名使用下划线连接单词，例如：USR\_CUSTOMER\_UID

## 源文件命名及编码

源文件使用大驼峰命名，使用UTF-8编码。

## 包命名规范

包名使用com.<group\_name>.<proj\_name>.<module>的命名方式。针对eOrder订餐系统，group\_name值为innovaee，proj\_name为eorder

|  |  |
| --- | --- |
| 顶级包名 | com.innovaee.eorder |
| 工具包 | com.innovaee.eorder.util |
| 配置包 | com.innovaee.eorder.conf |
| Model包 | com.innovaee.eorder.model |
| Service接口包 | com.innovaee.eorder.service |
| Service 实现包 | com.innovaee.eorder.service.imp |
| Controller包 | com.innovaee.eorder.controller |
| 异常包 | com.innovaee.eorder.exception |

## 类命名规范

类命名使用大驼峰命名，且必须采用名词。例如：UserInfo。接口定义最好采用名词，也可以采用形容词，例如：Readable。单元测试类采用相应类名加Test命名规则，例如：LoginServiceTest。

## 方法命名规范

方法名采用小驼峰命名，多采用动词和动词短语。例如：payBill()

## 常量命名规范

常量采用全大写希腊式命名，必须使用名词。例如：USR\_CUSTOMER\_UID

## 变量命名规范

变量命名使用小驼峰命名，使用名词。例如：currentDish

## 参数命名规范

参数命名采用小驼峰命名，使用名词。

# 源文件结构

## 项目目录结构

项目文件夹结构如下

## 包引入规范

import 语句不允许出现通配符，以避免引入未使用的类/接口。包引入根据提供方不同进行分组，且应遵循如下顺序

* static imports
* com.innovaee.eorder 命名空间下的引入
* 第三方包引入，例如Spring，Hibernate
* Java包引入
* Javax包引入

## 类声明规范

一个源文件只允许有一个类定义，如果因为编码需要引入内部类(inner-class)，需将内部类单独列出。对于匿名内部类不做要求。

## 注释规范

### 源文件注释

每个源文件在开始处应有关于文件版权声明，源文件名，创建日期等信息

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Filename : DishService.java

\* Copyright : Copyright (c) 2014

\* Company : Innovaee

\* Created : 11/27/2014

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

### 类注释

每个类名定义开始位置应有关于作者，类简单说明等信息

/\*\*

\*

\* Class : DishService<br/>

\* Description : Dish service interface definition<br/>

\* @author : LUO ZHI (kinglz2003@hotmail.com)<br/>

\* @version : 0.1 <br/>

\*/

### 变更注释

接着类注释下面应该有完整的变更历史，每条变更历史包括变更人，时间，引起变更的需求描述。

/\*\*

\* CR Number Date Author Description<br/>

\* ------------------------------------------------------------ --------------------------------------------------<br/>

\* CR987 11/27/2014 LUO ZHI This is an example of CR

\*/

### 方法注释

每个方法应有方法注释，包括方法作用，输入参数，输出参数等信息

/\*\*

\* This section to describe the usage of this method.

\*

\* @param param\_1 String the name of dish

\* @return the entity of a specific Dish

\*/

### 代码内注释

代码内注释用于解释比较复杂的业务逻辑，核心算法

### Javadoc

# 代码排版

## 括号

括号与if, else, for ,do, while等连接代码块的语句一起使用，规则如下，

* 左括号{之前没有换行符
* 左括号之后紧跟换行符
* 右括号之前换行

例如下代码：

*while (conditionIsTrue) {*

*}*

## 缩进

需要缩进的语句采用4字符缩进。

## 每行字符限制

单行代码字符不要超过80字符，如果超出应合理换行

## 单行语句

一个代码行只有一个语句

## 代码行限制

对于逻辑复杂的代码块或者方法，代码行除开括号应控制在60行以内。

# 代码书写规范

## 简单语句

## 复杂语句

## 返回语句

## 条件语句

## 分支语句

## 循环语句

## 异常控制

## 访问控制

# 日志规范

## 禁止使用System.out.println

## Log4j配置

## 日志等级

## 日志分类

## 日志格式

## 日志定义与继承

# 编程惯例

## 常量

## 异常按层次抛出

## 国际化

## 字符串生成

## Model类建议实现接口

## 面向接口编程

## 避免使用自带同步机制的集合类

## 变量初始化