目录

[一、命名规范](#_Toc23618_WPSOffice_Level1) [2](#_Toc23618_WPSOffice_Level1)

[1.1 命名可用字符](#_Toc27651_WPSOffice_Level2) [2](#_Toc27651_WPSOffice_Level2)

[1.2 包名](#_Toc22183_WPSOffice_Level2) [2](#_Toc22183_WPSOffice_Level2)

[1.2.1 个人](#_Toc27651_WPSOffice_Level3) [2](#_Toc27651_WPSOffice_Level3)

[1.2.2 团队](#_Toc22183_WPSOffice_Level3) [2](#_Toc22183_WPSOffice_Level3)

[1.2.3 公司](#_Toc28116_WPSOffice_Level3) [2](#_Toc28116_WPSOffice_Level3)

[1.3 类名](#_Toc28116_WPSOffice_Level2) [2](#_Toc28116_WPSOffice_Level2)

[1.4 方法名](#_Toc16877_WPSOffice_Level2) [3](#_Toc16877_WPSOffice_Level2)

[1.5 变量名](#_Toc23542_WPSOffice_Level2) [4](#_Toc23542_WPSOffice_Level2)

[1.6 常量名](#_Toc24912_WPSOffice_Level2) [4](#_Toc24912_WPSOffice_Level2)

[二、代码格式](#_Toc27651_WPSOffice_Level1) [4](#_Toc27651_WPSOffice_Level1)

[2.1 缩进](#_Toc21388_WPSOffice_Level2) [4](#_Toc21388_WPSOffice_Level2)

[2.2 行宽](#_Toc16365_WPSOffice_Level2) [4](#_Toc16365_WPSOffice_Level2)

[2.3 代码块格式](#_Toc5057_WPSOffice_Level2) [4](#_Toc5057_WPSOffice_Level2)

[2.4声明规范](#_Toc26318_WPSOffice_Level2) [5](#_Toc26318_WPSOffice_Level2)

[三、注释规范](#_Toc22183_WPSOffice_Level1) [5](#_Toc22183_WPSOffice_Level1)

[3.1 Java Doc](#_Toc30138_WPSOffice_Level2) [5](#_Toc30138_WPSOffice_Level2)

[3.2 块级别注释](#_Toc5546_WPSOffice_Level2) [5](#_Toc5546_WPSOffice_Level2)

一、命名规范

1.1 命名可用字符

所有的标识符都只能用ASCⅡ字母（A-Z或a-z）、数字（0-9）和下划线“\_”命名。

1.2 包名

一个唯一包名的前缀必须全部是小写的字母。Java包名可以自定义，但是经查阅资料得知，为了规范代码，便于合作开发以及后期维护，包名的命名一般遵循以下命名规则。

1.2.1 个人

indi：多人完成，版权属于发起者。包名为indi.发起者名.项目名.模块名......

pers：独自完成，公开，版权主要属于个人。包名为pers.个人名.项目名.模块名......

priv：独自完成，非公开，版权属于个人。包名为priv.个人名.项目名.模块名......

1.2.2 团队

team：团队项目指由团队发起，并由该团队开发的项目，版权属于该团队所有。包名为team.团队名.项目名.模块名......

1.2.3 公司

com：由公司发起，版权由项目发起的公司所有。包名为com.公司名.项目名.模块名......

1.3 类名

类名是一个名词，采用大小写混合的方式，每个单词的首字母大写。类名的不同后缀表示不同的效果。

|  |  |  |
| --- | --- | --- |
| 后缀名 | 意义 | 举例 |
| Service | 表明这个类是个服务类，里面包含了给其他类提同业务服务的方法 | PaymentOrderService |
| Impl | 这个类是一个实现类，而不是接口 | PaymentOrderServiceImpl |
| Inter | 这个类是一个接口 | LifeCycleInter |
| Dao | 这个类封装了数据访问方法 | PaymentOrderDao |
| Action | 直接处理页面请求，管理页面逻辑了类 | UpdateOrderListAction |
| Listener | 响应某种事件的类 | PaymentSuccessListener |
| Event | 这个类代表了某种事件 | PaymentSuccessEvent |
| Servlet | 一个Servlet | PaymentCallbackServlet |
| Factory | 生成某种对象工厂的类 | PaymentOrderFactory |
| Adapter | 用来连接某种以前不被支持的对象的类 | DatabaseLogAdapter |
| Job | 某种按时间运行的任务 | PaymentOrderCancelJob |
| Wrapper | 这是一个包装类，为了给某个类提供没有的能力 | SelectableOrderListWrapper |
| Bean | 这是一个POJO | MenuStateBean |

1.4 方法名

方法名是一个动词或动词词组，采用大小写混合的方式，第一个单词的首字母小写，其后单词的首字母大写，遵循驼峰命名法。方法名的不同动词前缀表示不同的含义。

|  |  |  |
| --- | --- | --- |
| 前缀名 | 意义 | 举例 |
| create | 创建 | createOrder() |
| delete | 删除 | deleteOrder() |
| add | 创建，暗示新创建的对象属于某个集合 | addPaidOrder() |
| remove | 删除 | removeOrder() |
| init或则initialize | 初始化，暗示会做些诸如获取资源等特殊动作 | initializeObjectPool |
| destroy | 销毁，暗示会做些诸如释放资源的特殊动作 | destroyObjectPool |
| open | 打开 | openConnection() |
| close | 关闭 | closeConnection()&lt; |
| read | 读取 | readUserName() |
| write | 写入 | writeUserName() |
| get | 获得 | getName() |
| set | 设置 | setName() |
| prepare | 准备 | prepareOrderList() |
| copy | 复制 | copyCustomerList() |
| modity | 修改 | modifyActualTotalAmount() |
| calculate | 数值计算 | calculateCommission() |
| do | 执行某个过程或流程 | doOrderCancelJob() |
| dispatch | 判断程序流程转向 | dispatchUserRequest() |
| start | 开始 | startOrderProcessing() |
| stop | 结束 | stopOrderProcessing() |
| send | 发送某个消息或事件 | sendOrderPaidMessage() |
| receive | 接受消息或时间 | receiveOrderPaidMessgae() |
| respond | 响应用户动作 | responseOrderListItemClicked() |
| find | 查找对象 | findNewSupplier() |
| update | 更新对象 | updateCommission() |

1.5 变量名

变量名第一个字母小写，任何中间单词的首字母大写。变量名应简短且可以顾名思义，易于记忆。例如：neekeName、neekeAddress。避免单个字符的变量名，除非是一次性的临时变量。

1.6 常量名

常量的声明应该全部大写，每个单词之间用“\_”连接。例如：final String HELLO\_WORLD = "hello world!";

二、代码格式

2.1 缩进

最好用空格字符缩进源代码，不要用tab，每次缩进4个空格。

2.2 行宽

每行字符不能超过60个，超出部分换行编写，例如字符串特别长，可以使用字符串的加操作，分成多行进行编写。

2.3 代码块格式

大括号的开始在代码块开始的行尾，闭合在和代码块同一缩进的行首，例如：

public class Test {

public static void main(String[] args) {

System.out.println("Hello World!");

}

}

即使if……语句只包含一条简单语句，也必须用“{”和“}”括起来。

2.4声明规范

一行声明一个变量。

不要将不同类型变量的声明放在同一行。

只在代码块的开始处声明变量。

所有的变量必须在声明时初始化。

避免声明的局部变量覆盖上一级声明的变量。

方法与方法之间以空行分隔。

三、注释规范

3.1 Java Doc

表明类、域和方法等的意义和用法等的注释，要以javadoc的方式来写。Java Doc是个类的使用者来看的，主要介绍 是什么，怎么用等信息。凡是类的使用者需要知道，都要用Java Doc 来写。非Java Doc的注释，往往是个代码的维护者看的，着重告述读者为什么这样写，如何修改，注意什么问题等。 如下：

/\*\*

This is a class comment

\*/

public class TestClass {

/\*\*

This is a field comment

\*/

public String name;

/\*\*

This is a method comment

\*/

public void call() {

}

}

3.2 块级别注释

块级别注释，单行时用 //, 多行时用 /\* … \*/。

较短的代码块用空行表示注释作用域，较长的代码块用/\*------ start: ------\*/和/\*-------- end: -------\*/包围。