Java语言代码规范

石家庄铁道大学信息科学与技术学院

信1705-1班

陈欢

目录

[Java语言代码规范 1](#_Toc38631605)

[1. 标识符命名规范 3](#_Toc38631606)

[1.1 统一、达意、简洁 3](#_Toc38631607)

[1.2 包名 3](#_Toc38631608)

[1.3 类名 3](#_Toc38631609)

[1.3.1 首字母大写 3](#_Toc38631610)

[1.3.2 后缀 3](#_Toc38631611)

[1.4 方法名 4](#_Toc38631612)

[2. 代码格式 5](#_Toc38631613)

[2.1 源文件编码 6](#_Toc38631614)

[2.2 行宽 6](#_Toc38631615)

[2.3 包的导入 6](#_Toc38631616)

[2.4 代码块格式 6](#_Toc38631617)

[2.4.1 缩进风格 6](#_Toc38631618)

[3. 注释规范 8](#_Toc38631619)

[3.1 Java Doc 8](#_Toc38631620)

[3.2 块级别注释 8](#_Toc38631621)

[3.2.1 块级别注释，单行时用 //, 多行时用 /\* … \*/。 9](#_Toc38631622)

[3.2.2 较短的代码块用空行表示注释作用域 9](#_Toc38631623)

[3.2.3 较长的代码块要用 9](#_Toc38631624)

# 1. 标识符命名规范

## 1.1 统一、达意、简洁

统一：有多个表达时，在一个项目中应选定一种；

达意：能够起到顾名思义的作用和效果；

简洁：在做到前两点的前提下，尽量简短。

## 1.2 包名

全小写，不隔断

## 1.3 类名

### 1.3.1 首字母大写

类名要首字母大写，兼顾驼峰命名

### 1.3.2 后缀

不同后缀实现表达不同意思的效果，如下表：

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

首字母小写，如 addOrder() 不要 AddOrder()

动词在前，如 addOrder()，不要orderAdd()

动词前缀往往表达特定的含义，如下表：

|  |  |  |
| --- | --- | --- |
| 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() |

find方法在业务层尽量表达业务含义，比如 findUnsettledOrders()，查询未结算订单，而不要findOrdersByStatus()。 数据访问层，find,update等方法可以表达要执行的sql，比如findByStatusAndSupplierIdOrderByName(Status.PAID, 345)

# 2. 代码格式

用空格字符缩进源代码，不要用tab，每个缩进4个空格。

## 2.1 源文件编码

源文件使用utf-8编码，结尾用unix n 分格。

## 2.2 行宽

行宽度不要超过130。

## 2.3 包的导入

删除不用的导入，尽量不要使用整个包的导入。在eclipse下经常使用快捷键 ctrl+shift+o 修正导入。

## 2.4 代码块格式

### 2.4.1 缩进风格

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：

package com.test;

public class TestStyle extends SomeClass implements AppleInter, BananaInter {

public static final String THIS\_IS\_CONST = "CONST VALUE";

private static void main(String[] args) {

int localVariable = 0;

}

public void compute(String arg) {

if (arg.length() &gt; 0) {

System.out.println(arg);

}

for (int i = 0; i &lt; 10; i++) {

System.out.println(arg);

}

while (condition) {

}

do {

otherMethod();

} while (condition);

switch (i) {

case 0:

callFunction();

break;

case 1:

callFunctionb();

break;

default:

break;

}

}

}

# 3. 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

### 3.2.1 块级别注释，单行时用 //, 多行时用 /\* … \*/。

### 3.2.2 较短的代码块用空行表示注释作用域

### 3.2.3 较长的代码块要用

/\*------ start: ------\*/

和

/\*-------- end: -------\*/

包围

如：

/\*----------start: 订单处理 ------- \*/

//取得dao

OrderDao dao = Factory.getDao("OrderDao");

/\* 查询订单 \*/

Order order = dao.findById(456);

//更新订单

order.setUserName("uu");

order.setPassword("pass");

order.setPrice("ddd");

orderDao.save(order);

/\*----------end: 订单处理 ------- \*/