**校园订餐系统代码开发规范**

**V0.1**

|  |  |  |  |
| --- | --- | --- | --- |
| 项目编号： |  | 项目名称： |  |
| 编 写 人： |  | 编写日期： |  |
| 审 核 人： |  | 审核日期： |  |
| 批 准 人： |  | 批准日期： |  |

**修订记录**

|  |  |  |
| --- | --- | --- |
| 修订人 | 修订日期 | 修订内容 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 1.引言

## 1.1编写目的

通过建立编码规范，以使每个开发人员养成良好的、统一的编码风格和习惯，提高程序的可靠性、可读性、可修改性、可维护性和一致性，保证软件产品的质量。

## 1.2适用范围

1）架构师

2）程序员

# 2.工具使用规范

## 2.1开发工具要求

|  |  |  |
| --- | --- | --- |
| 编号 | 软件工具 | 备注 |
| 1 | Eclipse |  |
| 2 | Tomcat8.5 |  |
| 3 | mysql |  |
| 4 | SUN JDK1.8 |  |
| 5 | Maven3.5.3 |  |

# 3.开发规范

## 3.1.java规范

## 3.1.1.命名

1.包的命名

Java包的名字都是由小写单词组成。但是由于Java面向对象编程的特性，每一名Java程序员都 可以编写属于自己的Java包，为了保障每个Java包命名的唯一性，在最新的Java编程规范中，要求程序员在自己定义的包的名称之前加上唯一的前缀。 由于互联网上的域名称是不会重复的，所以程序员一般采用自己在互联网上的域名称作为自己程序包的唯一前缀。

例如：edu.zut.cs.software.hero.\*\*\*

2. 类的命名

根据约定，Java类名通常以大写字母开头，如果类名称由多个单词组成，则每个单词的首字母均应为大 写例如UserManager；如果类名称中包含单词缩写，则这个所写词的每个字母均应大写，如：XMLExample,还有一点命名技巧就是由于类是设计用来 代表对象的，所以在命名类时应尽量选择名词。

例如：Food

3.方法的命名

方法的名字的第一个单词应以小写字母作为开头，后面的单词则用大写字母开头。

例如：findByUserName

4.参数的命名

参数的命名规范和方法的命名规范相同，而且为了避免阅读程序时造成迷惑，请在尽量保证参数名称为一个单词的情况下使参数的命名尽可能明确。

## 3.1.2格式

1. /\*\*
2. \* Java编码格式个人推荐，参考JDK源码和Hyperic HQ源码（原spring旗下著名开源软件，现vmware）。
3. \* @author zhangdfa
4. \*/
5. public class CodeRule {
7. //声明变量，等号两边有空格。
8. private static int i = 1;
10. //方法声明，右括号和左大括号中间有空格。
11. public static void main(String[] args) {
12. //if语句，比较连接符(>)左右有空格，小括号和大括号中间有空格。
13. //if 与 左括号中间有空格
14. if (i > 0) {
15. System.out.println(i);
16. }
17. //两个条件的连接(&&)，左右有空格。
18. if (i > 0 && i < 2) {
19. System.out.println(i);
20. }
22. //if..else 语句两种格式
23. //1.参考JDK，个人使用方式，else跟大括号，前后都有空格
24. if (i > 0 && i < 2) {
25. System.out.println(i);
26. } else if (i > 2) {
27. System.out.println(i + 1);
28. } else {
29. System.out.println(i);
30. }
31. //2.参考Hyperic HQ源码, else另起一行，后仍有空格
32. if (i == 1) {
33. System.out.println(i);
34. }
35. else {
36. System.out.println(i);
37. }
39. //while语句，与if语句类型，while与括号中间有空格，括号内格式与if相同
40. while (i > 0 && i < 2) {
41. System.out.println(i);
42. i++;
43. }
45. //for语句，两种格式
46. //1.参考Hyperic HQ，个人使用方式。分号后带空格，每个子语句中，连接符左右都带空格。
47. //for与括号中间带空格，大小括号中间带空格。
48. for (int j = 0; j < 10; j++) {
49. System.out.println(i);
50. }
51. //2.参考JDK，区别在于子语句中，连接符左右无空格。
52. for (int j=0; j<10; j++) {
53. System.out.println(i);
54. }
56. //+-\*/，格式，四则运算符号前后有空格。
57. //在JDK的有些代码里，在方法调用的参传递或在判断语句中存在的四则运算中，四则运算符号前后无空格。
58. //为了不造成困扰和混淆，个人为均保留空格。
59. int a = 1 + 2;
60. int b = 1 - 2;
61. int c = 1 \* 2;
62. int d = 1 / 2;
64. //三元表达式格式，每个符号中间均有空格
65. int j = i > 2 ? 1 : -1;
67. //方法声明和调用，用逗号分隔的参数，逗号后有空格。
68. sum(a, b);
69. sum(c + d, j);
70. }
72. //方法声明，多个参数，逗号后有空格
73. private static int sum(int i, int j) {
74. return i + j;
75. }
76. }

# 4.打包及jar包升级

## 4.1基本原则

通常规则：总版本号.分支版本号.小版本号-里程碑版本

总版本号的变动一般表示框架的变动

分支版本号一般表示增加了一些功能

小版本号在分支版本上面进行bug的修改

## 4.2打包

### 4.2.1. 基于框架的rar包

一般为：项目名称-功能名称.rar,小版本号一般可以忽略

如：

