

后端编码规范

五成胜算队

- [1. 基本原则](#)
- [2. 类编写规范](#)
- [3. 方法编写规范](#)
- [4. 变量编写规范](#)
- [5. 参考资料](#)

1. 基本原则：

(1) 源文件编码格式为UTF-8。

(2) 用两个空格来代替制表符（tab）-- 这是唯一能保证在所有环境下获得一致展现的方法.且嵌套元素应当缩进一次（即两个空格）。

```
when(i != 0) {  
    break;  
}
```

(3) 为了代码的易读性，在每个方法块或者类块的左花括号前添加一个空格。声明块的右花括号应当单独成行。

```
class Student {  
    private int number  
    ...  
}
```

(4) 对于以逗号分隔的属性值，每个逗号后面都应该插入一个空格（例如，box-shadow）。

(5) 基本的风格是操作符的两边都要用空格，如：

```
a = b * 2 + c;
```

但++/--操作与表达式之间不要用空格，如应使用：

```
i++;  
i--;
```

(6) 让一切东西都尽可能地"私有"--private。可使库的某一部分"公共化"（一个方法、类或者一个字段等等），就永远不能把它拿出。若强行拿出，就可能破坏其他人现有的代码，使他们不得不重新编写和设计。若只公布自己必须公布的，就可放心大胆地改变其他任何东西。在多线程环境中，隐私是特别重要的一个因素--只有private字段才能在非同步使用的情况下受到保护。

(7) 如果不是必须，不要在循环中去用try...catch.

(8) 类中对于比较复杂的逻辑要采用块注释（`/**/`）的方式进行注释，简单的逻辑或者变量可以使用行注释（`//`）的方式进行注释。

```
class Table {  
    ...  
    private void sortTable {    //bibibibibibibi , 这里是.....  
        ...  
    }  
  
    /*  
    bibibibibibibi , 这里是.....  
    */  
    private void setFake {  
        ...  
    }  
}
```

(9) 注解紧跟在文档块后面，应用于类、方法和构造函数，一个注解独占一行。这些换行不属于自动换行(第4.5节，自动换行)，因此缩进级别不变。例如：

```
@Override@Nullablepublic String getNameIfPresent() { ... }
```

例外：单个的注解可以和签名的第一行出现在同一行。例如：

```
@Override public int hashCode() { ... }
```

应用于字段的注解紧随文档块出现，应用于字段的多个注解允许与字段出现在同一行。例如：

```
@Partial @Mock DataLoader loader;
```

参数和局部变量注解没有特定规则。

2. 类编写规范：

(1) 类名首字母大写

```
class Student {  
    ...  
}
```

(2) 对于设计期间不需要继承的类，尽量使用final。

(3) 类声明时，要指出其访问控制，一般为没有修饰符，public，protected和private。

(4) 编写通用性的类时，遵守标准形式。包括定义equals()、hashCode()、toString()、Clone(实现Cloneable接口)，并实现Comparable和Serializable接口。

3. 方法编写规范：

(1) 对成员方法,不要轻易的采用public的成员变量。主要的修饰符有public, private, protected。

```
class Student {  
    public void sortNumber() {...}  
    private Spring sdsDkj() {...}  
    ...  
}
```

(2) 方法和方法之间空一行

(3) 方法命名使用小驼峰法，例如：

```
public void firstNum() {...}
```

(4) 方法的文档注释放在方法的紧前面，不能空一行。

(5) 应将方法设计成简要的、功能性单元，用它描述和实现一个不连续的类接口部分。理想情况下，方法应简明扼要。若长度很大，可考虑通过某种方式将其分割成较短的几个方法。这样做也便于类内代码的重复使用（有些时候，方法必须非常大，但它们仍应只做同样的一件事情）。

4. 变量编写规范：

(1) 对成员变量, 尽量采用private。

(2) 每一个变量声明/定义占一行（参数变量除外），如：

```
int a;  
int b;
```

(3) 局部变量在使用前必须初始化，一般在声明时初始化。

(4) 常量统一使用全大写，单词之间用_连接，如:

```
static final int NTSE_NAME_LEN;
```

(5) 变量的声明要放在程序块的开始位置,如：

```
public void myMethod() {  
    int int1 = 0;          // beginning of method block  
    if (condition) {  
        int int2 = 0;      // beginning of "if" block  
        ...  
    }  
}
```

5.参考资料

- 《构建之法》第四章
- [【java开发规范总结_代码编码规范】](#)