# 编码规范

1. 每一个Class以及method之上，都需要编写JavaDoc。

-JavaDoc的内容需要明确【何が何をする】、这是一个类/方法，它的目的是什么，或是保有什么。

1. 起名，起名要求简单扼要，看名知其意。不能有多余的单词或者意义不明的形容。

-如：entity包下的，就应该叫xxxEntity；

service包下的就应该叫xxxxService；

controller就叫xxxxController。

1. 新建类、方法等的操作，应该遵循，**如无必要，勿添新枝**的设计思想，应该思考是否有必要新建Controller，Service等等，在积存的类中添加一些功能是否合适，如果不合适，才考虑新建。

4.实体类包，统一叫成entity。如果是与数据库表配对的实体类，需要在注释里说明配对的数据库表明。

JavaDoc 是 Java 语言的官方文档生成工具，它从 Java 源代码中读取注释，并生成一组以 HTML 文件格式表示的 API 文档。通过使用 JavaDoc，开发人员可以为自己的代码写文档，并以一种标准化的方式与其他开发人员分享这些文档。

JavaDoc 注释有其特定的格式。它们以 `/\*\*` 开始，并以 `\*/` 结束。在这之间，您可以使用一系列的特定标签来格式化和详细说明您的注释。以下是一个简单的例子来说明 JavaDoc 的注释：

事例代码

/\*\*

\* 代表一个简单的矩形。

\* <p>

\* 这个类提供了计算矩形面积和周长的方法。

\* </p>

\*

\* @author 张三

\* @version 1.0

\*/

public class Rectangle {

private double width;

private double height;

/\*\*

\* 创建一个新的矩形实例。

\*

\* @param width 矩形的宽度

\* @param height 矩形的高度

\*/

public Rectangle(double width, double height) {

this.width = width;

this.height = height;

}

/\*\*

\* 计算并返回矩形的面积。

\*

\* @return 矩形的面积

\*/

public double getArea() {

return width \* height;

}

// ... 其他方法和文档 ...

}

在上述代码中，我们使用了一些常见的 JavaDoc 标签，如 `@author` 和 `@version`，以及方法级别的 `@param` 和 `@return` 标签。

一旦你为代码添加了 JavaDoc 注释，你可以使用 `javadoc` 工具来生成 HTML 格式的 API 文档。这些生成的文档对于其他开发人员了解和使用你的代码非常有帮助，因为它们提供了一个清晰、易读的参考资料。

总之，JavaDoc 是一个强大的工具，用于为 Java 代码生成官方风格的 API 文档。正确和一致地使用它可以使你的代码更加易于理解和使用。