

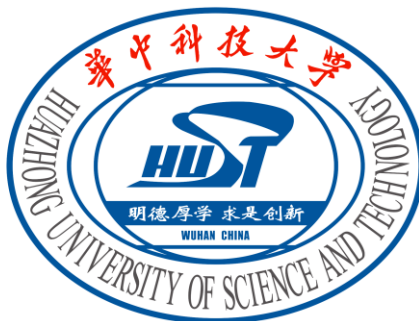
基于Java的面向对象程序设计

陈维亚

weiya_chen@hust.edu.cn

华中科技大学软件学院

第29讲：代码规范化之路



1. Coding Style

2. Javadoc

□ 总览

排版

命名

代码编码

中英文输入法

□ 一般原则

- **保持原有风格**
- **坚持最小惊奇原则**
 - **简单性**：构建简单的类和方法。
 - **清晰性**：明确每个变量、类、接口、方法、对象的用途。
 - **完整性**：提供用户所期望的完整的程序功能，并且为此编写完整的说明文档。
 - **一致性**：相似的实体，其外观和行为也要大致相同；反之亦然。
 - **健壮性**：对错误和异常做出预测，并且在说明文档中提供相应的解决办法。
- **第一次就做对**
- **记录所有非规范行为**

1. Coding Style



□ 排版

项目级

文件级

代码级

□ 排版 - 项目级

- 每个文件只包含一个类（内部类除外）
- 文件名和类名相同
- 大小写敏感，文件扩展名为.java
- 文件按功能模块进行组织，分放在不同的package之中

□ 排版 - 文件级

源文件编码格式为UTF-8

import少用通配符：import java.util.*;

一个源文件包含：

- 1) 许可证或版权信息
- 2) package语句
- 3) import语句
- 4) 类定义

以上每个部分之间用一个空行隔开。

```
/*
 * Alessandro Martinelli's Jopl Tutorial
 * Copyright (C) 2008 Alessandro Martinelli <alessandro.martinelli@unipv.it>
 *
 * This file is part of Alessandro Martinelli's Jopl Tutorial.
 *
 * Alessandro Martinelli's Jopl Tutorial is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
 */
```

```
package tutorials5;
```

```
import java.awt.image.BufferedImage;
```

```
import java.io.File;
```

```
import javax.swing.JFrame;
```

```
public class Practice01 extends JFrame{...}
```

□ 排版 - 代码级

类的声明和定义

1) 类成员顺序

类的成员顺序对易学性有很大的影响，虽不存在唯一的通用法则，但应该以某种逻辑去维护成员的排列顺序。

成员变量在成员方法之前，public 和 private 的成员要分开。

2) 重载方法

当一个类有多个构造函数，或是多个同名方法，这些方法中间不要放进其它方法。

□ 排版 - 代码级

代码块（大括号的使用）

- 1) 大括号与 if, else, for, do, while 语句一起使用，即使只有一条语句(或是空)，也应该把大括号写上。
- 2) 左大括号前不换行，后换行
- 3) 右大括号前换行，右大括号后某些情况不换行。
- 4) 一行只有一条语句，每行不宜过长（通常限制80或100字符）。
- 5) 正确使用缩进和空行（空白），慎用 Tab 键。

```
return new MyClass() {  
    @Override public void method() {  
        if (condition()) {  
            try {  
                something();  
            } catch (ProblemException e) {  
                recover();  
            }  
        }  
    }  
};
```

```
void doNothing() {}
```

□ 排版 - 代码级

一些细节

- 1) 每次只声明一个变量，避免如 `int a, b;`
- 2) 需要时才声明，不用像C语言一样提前声明
- 3) 中括号是类型的一部分：`String[] args`，而非`String args[]`。
- 4) 注解用于类、方法和构造函数，一个注解独占一行，因此缩进级别不变。

```
@Override
@Nullable
public String getNameIfPresent()
{ ... }
```

□ 命名规则

通用规则

标识符只能使用ASCII字母和数字

包名

包名全部小写，连续的单词只是简单地连接起来，不使用下划线。

类名 / 接口名

- 类名都以UpperCamelCase风格编写。
- 类名通常是名词或名词短语，接口名称有时可能是形容词或形容词短语。
- 测试类的命名以它要测试的类的名称开始，以Test结束。例如，HashTest 或 HashIntegrationTest。

□ 命名规则

方法名

方法名都以lowerCamelCase风格编写。

方法名通常是动词或动词短语。

常量名

常量名命名模式为CONSTANT_CASE，全部字母大写，用下划线分隔单词。

变量名

- 非常量字段名以lowerCamelCase风格编写，通常使用名词或名词短语。
- 参数名以lowerCamelCase风格编写，应该避免用单个字符命名。
- 局部变量名以lowerCamelCase风格编写，编写规则较为宽松。

□ 基本形式

按照标准化的注释结构，Javadoc工具可方便地生成HTML文档以供阅读。

```
/**  
 * Multiple lines of Javadoc text are written here,  
 * wrapped normally...  
 */  
public int method(String p1) { ... }
```

```
/** An especially short bit of Javadoc. */
```

至少在每个public类及它的每个public和protected成员处使用Javadoc

□ 规范

javadoc 标记有如下一些：

@author	标明开发该类模块的作者
@version	标明该类模块的版本
@see	参考转向，也就是相关主题
@param	对方法中某参数的说明
@return	对方法返回值的说明
@exception	对方法可能抛出的异常进行说明

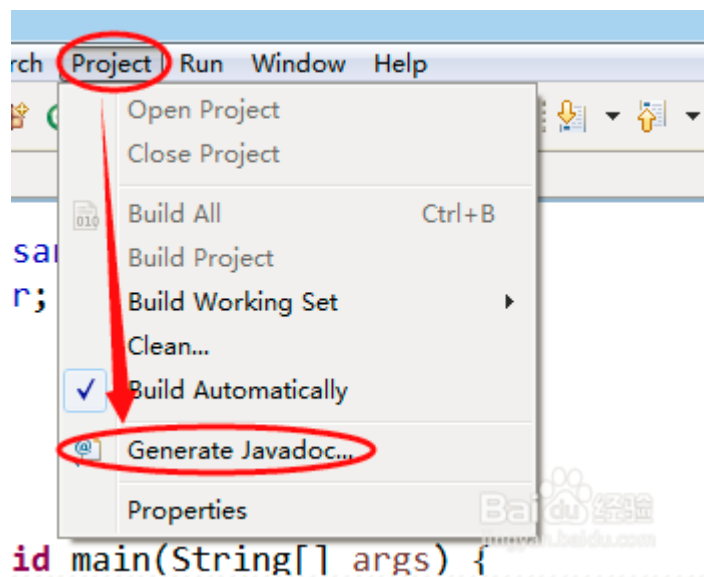
```
/**
 * 比较两个整数的大小并返回较大的整数。
 *
 * @param a 第一个整数
 * @param b 第二个整数
 * @return 两个参数中较大的整数
 */
public int max(int a, int b) {
    ...
}
```

□ 生成文档

- 命令行调用Javadoc

javadoc SquareNum.java

- Eclipse下的调用



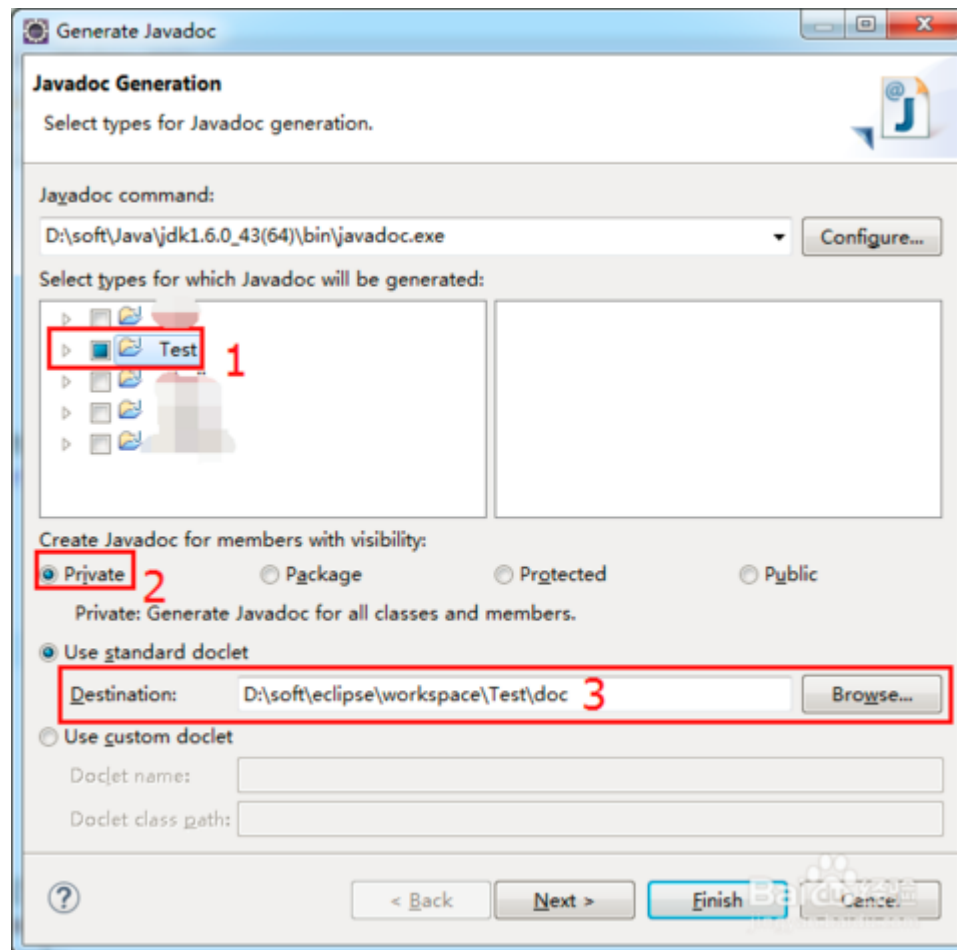
2. Javadoc

□ 生成文档

- 命令行调用Javadoc

javadoc SquareNum.java

- Eclipse下的调用



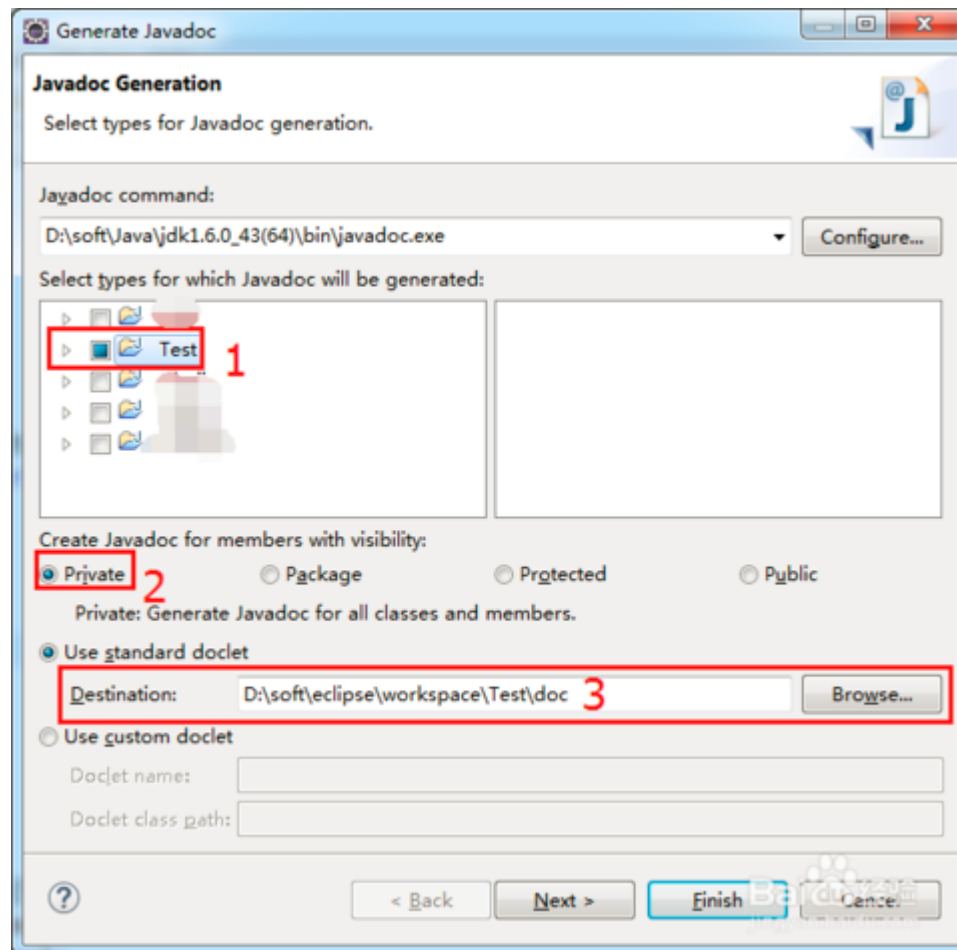
2. Javadoc

□ 生成文档

- 命令行调用Javadoc

javadoc SquareNum.java

- Eclipse下的调用

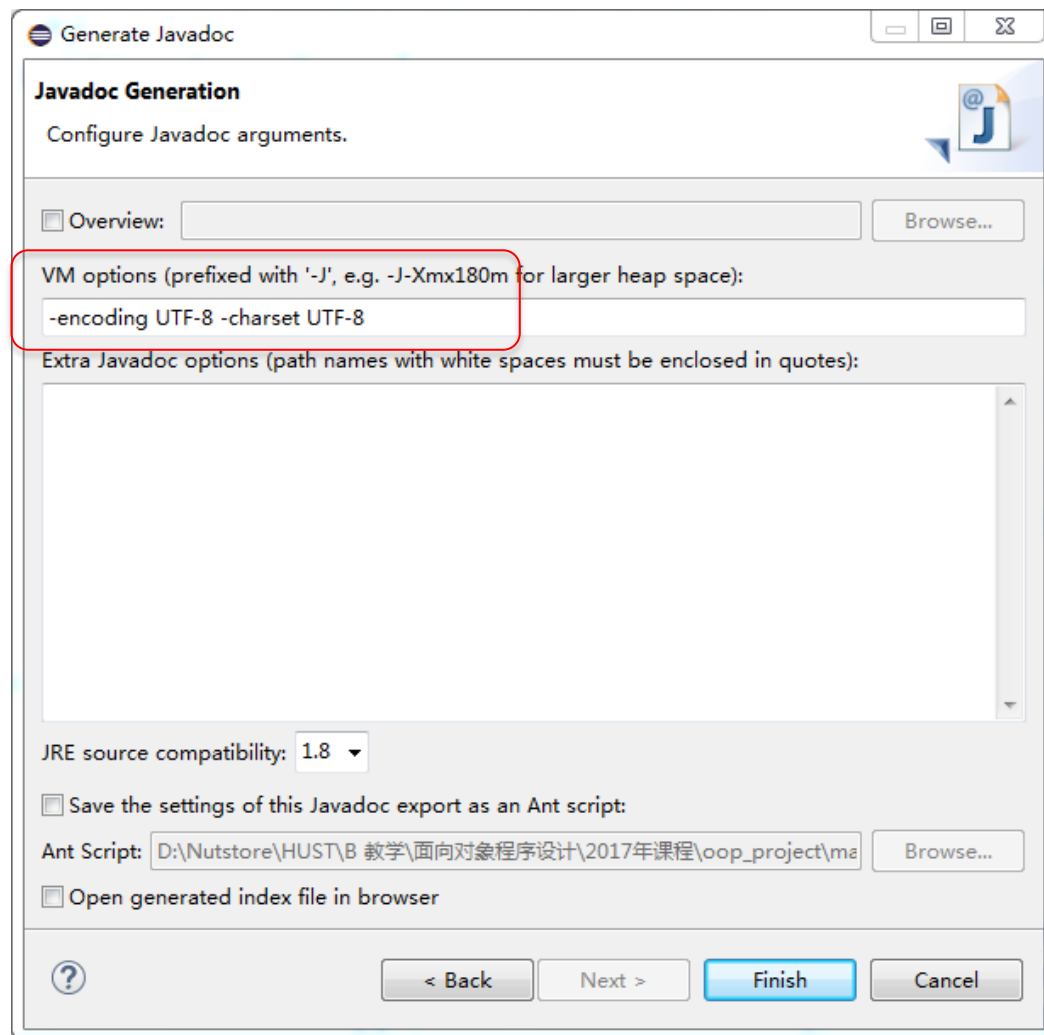


□ 生成文档

- 命令行调用Javadoc

`javadoc SquareNum.java`

- Eclipse下的调用



代码要规范

注释要详尽

好习惯改不了，坏习惯也改不了

课程设计指导课 2