Lab4 注释与Javadoc

- 注释
- Javadoc

刘诺纬

10191900446@stu.ecnu.edu.cn

注释

Java中,注释有两种方式,分别是单行注释和多行注释

单行注释:

```
public String helloWorld() {
    System.out.println("Hello, world!"); // 打印字符串Hello, world!
}
```

多行注释:

```
public String helloWorld() {
    /*
    打印字符串Hello, world!
    这是一个多行注释
    */
    System.out.println("Hello, world!");
}
```

Javadoc(最初是JavaDoc)是由Sun Microsystems为Java语言(现在由甲骨文公司拥有)创建的文档生成器,用于从Java源代码生成 HTML格式的API文档,HTML格式用于增加将相关文档链接在一起的便利性。

Javadoc使用的"doc comments"格式是记录Java类的事实上的行业标准。一些IDE,如IntelliJ IDEA、NetBeans和Eclipse,可以自动生成Javadoc HTML。许多文件编辑器帮助用户生成Javadoc源代码并使用Javadoc信息作为程序员的内部引用。

Javadoc还提供了用于创建doclet和taglet的API,允许用户分析Java应用程序的结构,这就是JDiff如何生成两个API版本之间发生变化的报告。

Javadoc不影响Java中的性能,因为在编译时会删除所有注释。编写注释和Javadoc是为了更好地理解代码,从而更好地维护代码。

节选自Wikipedia Javadoc词条

示例:

```
/**

* Statistic类用于计算常用的统计量,

* 包括最大值、最小值、均值、方差、第k大的数和统计数字在数组中出现的次数

*/
public class Statistic {
}
```

```
/**

* Statistic类用于计算常用的统计量,

* 包括最大值、最小值、均值、方差、第k大的数和统计数字在数组中出现的次数

*/
public class Statistic {
}
```

使用 `/**` 和 `*/` 来标记Javadoc的开始和结束,同时每行都有一个 `*` 开始。

再比如,加入一些作者、版本信息

```
/**

* Statistic类用于计算常用的统计量,

* 包括最大值、最小值、均值、方差、第k大的数和统计数字在数组中出现的次数

* @author 刘诺纬

* @version 1.0

*/
public class Statistic {
```

除了给类添加Javadoc之外,还可以给方法添加Javadoc。例如,希望给 `max() `添加Javadoc,说明该方法的作用、传入的参数以及返回的结果。

```
public class Statistic {
    /**
    * 计算数组中的最大值
    * @param a 数组
    * @return 数组中的最大值
    */
    public static double max(double[] a) {}
}
```

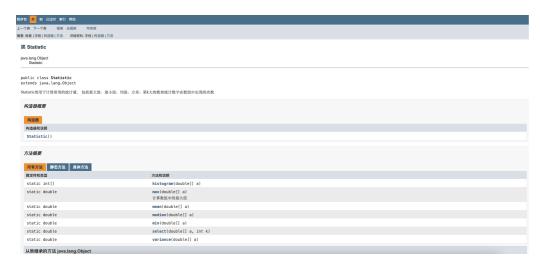
除此之外,Javadoc也支持HTML标记,例如使用``来表示段落

```
public class Statistic {
    /**
    * 计算数组中的最大值
    * @param a 数组
    * @return 数组中的最大值
    */
    public static double max(double[] a) {}
}
```

```
javadoc Statistic.java
```

在命令行中使用`javadoc filename`可以对某个文件或包生成Javadoc的API文档(filename.html即为生成出的静态页面,可以直接在浏览器中渲染)

当注释中有中文时,需要加上`-encoding utf8 -docencoding utf8 -charset utf8`表示使用utf-8字符集编码,这样渲染出的文档不会出现乱码。



这里列出常用的一些tag:

• @author: 作者的姓名

• @version: 版本号

• @param: 对于参数的描述

• @return: 对于返回内容的描述

• @exception (和@throws是同义词): 异常类的名称和描述

• @see:表示去查看参考资料

• @since:表示这个变更或特性从什么时候或版本号等(由该标签中声明的内容决定)开始存在的

• @serial: 用于表示序列化的字段 (include | exclude)

• @deprecated:表示被弃用

• @link: 用法{@link package.class#member label}。插入一个带标签的链接,可以指向特定包、类或指定类的成员名称的文档。

• @literal: 用法{@literal *text*}。用来显示那些不用被HTML标记或嵌套javadoc标签解析的文本。