Java Applet 基础 Java 实例 – 如何编译 Java 文件

Java 文档注释

Java 支持三种注释方式。前两种分别是 // 和 /* */，第三种被称作说明注释，它以 /** 开始，以 */结束。

说明注释允许你在程序中嵌入关于程序的信息。你可以使用 javadoc 工具软件来生成信息，并输出到HTML文件中。

说明注释，使你更加方便的记录你的程序信息。

javadoc 标签

javadoc 工具软件识别以下标签：

标签	描述	示例
@author	标识一个类的作者	@author description
@deprecated	指名一个过期的类或成员	@deprecated description
{@docRoot}	指明当前文档根目录的路径	Directory Path
@exception	标志一个类抛出的异常	@exception exception-name explanation
{@inheritDoc}	从直接父类继承的注释	Inherits a comment from the immediate surperclass.
{@link}	插入一个到另一个主题的链接	{@link name text}
{@linkplain}	插入一个到另一个主题的链接，但是该链接显示纯文本字体	Inserts an in-line link to another topic.
@param	说明一个方法的参数	@param parameter-name explanation
@return	说明返回值类型	@return explanation
@see	指定一个到另一个主题的链接	@see anchor
@serial	说明一个序列化属性	@serial description
@serialData	说明通过writeObject( ) 和 writeExternal( )方法写的数据	@serialData description
@serialField	说明一个ObjectStreamField组件	@serialField name type description
@since	标记当引入一个特定的变化时	@since release
@throws	和 @exception标签一样.	The @throws tag has the same meaning as the @exception tag.

{@value}	显示常量的值，该常量必须是static属性。	Displays the value of a constant, which must be a static field.
@version	指定类的版本	@version info

文档注释

在开始的 /** 之后，第一行或几行是关于类、变量和方法的主要描述。

之后，你可以包含一个或多个何种各样的 @ 标签。每一个 @ 标签必须在一个新行的开始或者在一行的开始紧跟星号(*).

多个相同类型的标签应该放成一组。例如，如果你有三个 **@see** 标签，可以将它们一个接一个的放在一起。

下面是一个类的说明注释的实例：

```
/*** 这个类绘制一个条形图
* @author runoob
* @version 1.2
*/
```

javadoc 输出什么

javadoc 工具将你 Java 程序的源代码作为输入，输出一些包含你程序注释的HTML文件。

每一个类的信息将在独自的HTML文件里。javadoc 也可以输出继承的树形结构和索引。

由于 javadoc 的实现不同，工作也可能不同，你需要检查你的 Java 开发系统的版本等细节，选择合适的 Javadoc 版本。

实例

下面是一个使用说明注释的简单实例。注意每一个注释都在它描述的项目的前面。

在经过 javadoc 处理之后，SquareNum 类的注释将在 SquareNum.html 中找到。

SquareNum.java 文件代码：

```
import java.io.*;
/**
* 这个类演示了文档注释
* @author Ayan Amhed
* @version 1.2
*/
public class SquareNum {
/**
* This method returns the square of num.
* This is a multiline description. You can use
* as many lines as you like.
* @param num The value to be squared.
* @return num squared.
*/
public double square(double num) {
return num * num;
}
/**
* This method inputs a number from the user.
* @return The value input as a double.
```

```
* @exception IOException On input error.
* @see IOException
*/
public double getNumber() throws IOException {
InputStreamReader isr = new InputStreamReader(System.in);
BufferedReader inData = new BufferedReader(isr);
String str;
str = inData.readLine();
return (new Double(str)).doubleValue();
}
/**
* This method demonstrates square().
* @param args Unused.
* @return Nothing.
* @exception IOException On input error.
* @see IOException
*/
public static void main(String args[]) throws IOException
{
SquareNum ob = new SquareNum();
double val;
System.out.println("Enter value to be squared: ");
val = ob.getNumber();
val = ob.square(val);
System.out.println("Squared value is " + val);
}
}
```

如下，使用 javadoc 工具处理 SquareNum.java 文件：

```
$ javadoc SquareNum.java
Loading source file SquareNum.java...
Constructing Javadoc information...
Standard Doclet version 1.5.0_13
Building tree for all the packages and classes...
Generating SquareNum.html...
SquareNum.java:39: warning - @return tag cannot be used\
                      in method with void return type.
Generating package-frame.html...
Generating package-summary.html...
Generating package-tree.html...
Generating constant-values.html...
Building index for all the packages and classes...
Generating overview-tree.html...
Generating index-all.html...
Generating deprecated-list.html...
Building index for all classes...
Generating allclasses-frame.html...
Generating allclasses-noframe.html...
Generating index.html...
Generating help-doc.html...
```

```
Generating stylesheet.css...
1 warning
$
```

← Java Applet 基础 Java 实例 – 如何编译 Java 文件 →

1 篇笔记

写笔记

看到这里觉得可以分享一下自己的注释，其实很多注释是可以自定义的。

定义成模板在自己的 IDE 上，这样每次通过快捷键就可自动帮你输出在方法中，省去了很多时间，也使代码更加规范。

下面已 eclipse 为例，分析一下自己的。

我是加载了 JAutodoc 插件在 IDE 中，习惯这种格式的小伙伴也可以去下载一下。

首先是在文件头部添加：

```
/*
 * <p>项目名称: ${project_name} </p>
 * <p>文件名称: ${file_name} </p>
 * <p>描述: [类型描述] </p>
 * <p>创建时间: ${date} </p>
 * <p>公司信息: ************公司 *********部</p>
 * @author <a href="mail to: *******@******.com" rel="nofollow">作者</a>
 * @version v1.0
 * @update [序号][日期YYYY-MM-DD] [更改人姓名][变更描述]
 */
```

方法：

```
/**
 * @Title: ${enclosing_method}
 * @Description: [功能描述]
 * @Param: ${tags}
 * @Return: ${return_type}
 * @author <a href="mail to: *******@******.com" rel="nofollow">作者</a>
 * @CreateDate: ${date} ${time}</p>
 * @update: [序号][日期YYYY-MM-DD] [更改人姓名][变更描述]
 */
```

getter 和 setter

```
/**
 * 获取  ${bare_field_name}
 */
```

```
/**
 * 设置   ${bare_field_name}
 * (${param})${field}
 */
```

JeoSaber 8个月前 (07-26)