**Doxygen 注释语法和使用**

目录

Doxygen注释格式

块注释

2. 文件注释

3. 类定义注释

4. 常量/变量的注释

5. 函数注释

其他

按照类型来注释

1）文件注释

2）函数注释

3）类型/宏定义注释

4）枚举/结构注释

Doxygen注释格式

块注释

建议统一使用



行注释建议统一使用



由于Doxygen 对于批注是视为在解释后面的程序代码。（以上默认在解释后面的程序）

例如:



针对一些常用的指令做说明：

@file

档案的批注说明。

@author

作者的信息

@brief

用于class 或function的批注中，后面为class 或function的简易说明。

@param

格式为

@param arg\_name 参数说明

主要用于函式说明中，后面接参数的名字，然后再接关于该参数的说明。

@return

后面接函数传回值的说明。用于function的批注中。说明该函数的传回值。

@retval

格式为

@retval value 传回值说明

主要用于函式说明中，说明特定传回值的意义。所以后面要先接一个传回值。然后在放该传回值的说明。

如果要批注前面的程序代码则需用下面格式的批注符号。

/\*!< ... 批注 ... \*/

/\*\*< ... 批注 ... \*/ （推荐）

//!< ... 批注 ...

///< ... 批注 ...  （推荐）

例如：



2. 文件注释

文件注释通常放在整个文件开头。



例如：



生成文档效果

3. 类定义注释

类定义的注释方式非常简单，使用@brief后面填写类的概述，换行填写类的详细信息。

/\*\*

\* @brief 类的简单概述

\* 类的详细概述

\*/

例如：

/\*\*

\* @brief 测试类

\* 主要用来演示Doxygen类的注释方式

\*/

class Test{

};

生成文档效果

命名空间、结构体、联合体、枚举定义与类定义注释方式一致。

4. 常量/变量的注释

常量/变量包括以下几种类型

全局常量变量

宏定义

类/结构体/联合体的成员变量

枚举类型的成员

注释分为两种方式，可根据具体情况自行选择

代码前注释

/// 注释

常量/变量

例如：

/// 缓存大小

#define BUFSIZ 1024\*4

代码后注释

常量/变量 ///< 注释

例如：

#define BUFSIZ 1024\*4 ///< 缓存大小

生成文档效果

5. 函数注释

简约注释

函数注释主要包含函数简介(@brief)、参数说明('@param')、返回说明(@return)和返回值说明(@retval)四部分。

/\*\*

\* @brief 函数简介

\*

\* @param 形参 参数说明

\* @param 形参 参数说明

\* @return 返回说明

\* @retval 返回值说明

\*/

详细注释

可以根据需要添加详细说明(@detail)、注解(@note)、注意(@attention)、警告(@warning)或者异常(@exception)等。



例子

以main()函数为例添加函数注释。



生成文档效果



其他

下面一些标注方式可以根据需要选择使用。

命令 生成字段名 说明

@see 参考

@class 引用类 用于文档生成连接

@var 引用变量 用于文档生成连接

@enum 引用枚举 用于文档生成连接

@code 代码块开始 与@endcode成对使用

@endcode 代码块结束 与@code成对使用

@bug 缺陷 链接到所有缺陷汇总的缺陷列表

@todo TODO 链接到所有TODO 汇总的TODO 列表

@example 使用例子说明

@remarks 备注说明

@pre 函数前置条件

@deprecated 函数过时说明

按照类型来注释

1）文件注释



2）函数注释



3）类型/宏定义注释



4）枚举/结构注释

/\*\* 枚举类型\*/

