

Doxygen注释语法---生成源码文档

语法简介

Doxygen 是一个程序的文件产生工具，可将程序中的特定批注转换成为说明文件。提供了一套注释方式便于把代码中的注释生成说明文档。让代码与文档同步进行，IBM公司用于AIX与UNIX的开发。

下面是常用的注释简介。

1. 简单注释

- 单行注释: `///` 或者 `//!`
- 多行注释: `/**` 或者 `/*!`

2. 文件注释

文件注释通常放在整个文件开头。

```
/**
 * @file 文件名
 * @brief 简介
 * @details 细节
 * @mainpage 工程概览
 * @author 作者
 * @email 邮箱
 * @version 版本号
 * @date 年-月-日
 * @license 版权
 */
```

例如：

```
/**
 * @file Test.h
 * @brief 测试头文件
 * @details 这个是测试Doxygen
 * @mainpage 工程概览
 * @author Loodao
 * @email Loodao@163.com
 * @version 1.0.0
 * @date 2020-04-01
 */
```

生成文档效果

工程概览

作者

jdzhangxin jdzhangxin@126.com

版本

1.0.0

日期

2017-11-17

制作者  1.8.13

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

类

```
class Test
    测试类 主要用来演示Doxygen类的注释方式 更多...
```

Test类 参考

测试类 主要用来演示Doxygen类的注释方式 [更多...](#)

```
#include <Test.h>
```

详细描述

测试类 主要用来演示Doxygen类的注释方式

该类的文档由以下文件生成:

- [Test.h](#)

制作者  1.8.13

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

```
#define BUFSIZ 1024*4  
缓存大小
```

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。

```
/**  
 * @brief 函数简介  
 * @detail 详细说明  
 *  
 * @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  
 * @note 注解  
 * @attention 注意  
 * @warning 警告  
 * @exception 异常  
 */
```

- 例子

以main()函数为例添加函数注释。

```

/**
 * @brief 主函数
 * @details 程序唯一入口
 *
 * @param argc 命令参数个数
 * @param argv 命令参数指针数组
 * @return 程序执行成功与否
 *      @retval 0 程序执行成功
 *      @retval 1 程序执行失败
 * @note 这里只是一个简单的例子
 */
int main(int argc, char* argv[])

```

生成文档效果

函数说明

◆ main()

```

int main ( int    argc,
           char * argv[]
         )

```

主函数

程序唯一入口

参数

argc 命令参数个数

argv 命令参数指针数组

返回

程序执行成功与否

返回值

0 程序执行成功

1 程序执行失败

注解

这里只是一个简单的例子

制作者  1.8.13

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	

参考资料：

<https://www.ibm.com/developerworks/cn/aix/library/au-learningdoxygen/#list2%20CommentManual.pdf>