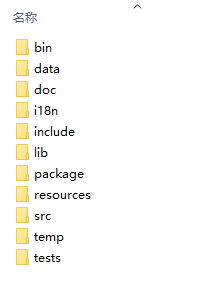
Windows下开发规范

文档规定c++ for vc开发规范，包括文件组织、程序注释、命名规则、函数及语句、其他规范。

# 文件组织

## 组织目录



### bin

《Bin》目录存放所有工程生成和第三方的dll、exe文件

### data

《Data》目录存放数据文件，一般会放一些图像以及训练好的模型文件

### doc

《Doc》目录存放一些帮助和说明文件、过程评审技术文档以及过程设计成果

### i18n

《I18n》国际化文件夹，存放语言包

### include

《Include》目录存放所有依赖的第三方头文件

### lib

《Lib》目录存放所有工程生成和所有第三方的lib文件,《Lib》目录视情况可以再分为《Debug》及《Release》

### package

代码工程依赖的三方库及安装包

系统资源文件夹，存放图片及图标等资源文件

### resources

系统资源文件夹，存放图片及图标等资源文件

### src

《Src》目录存放所有工程目录和.sln解决方案文件

### temp

《Temp》目录存放项目运行生成的一些中间文件，包括compile和link

### tests

测试文件夹

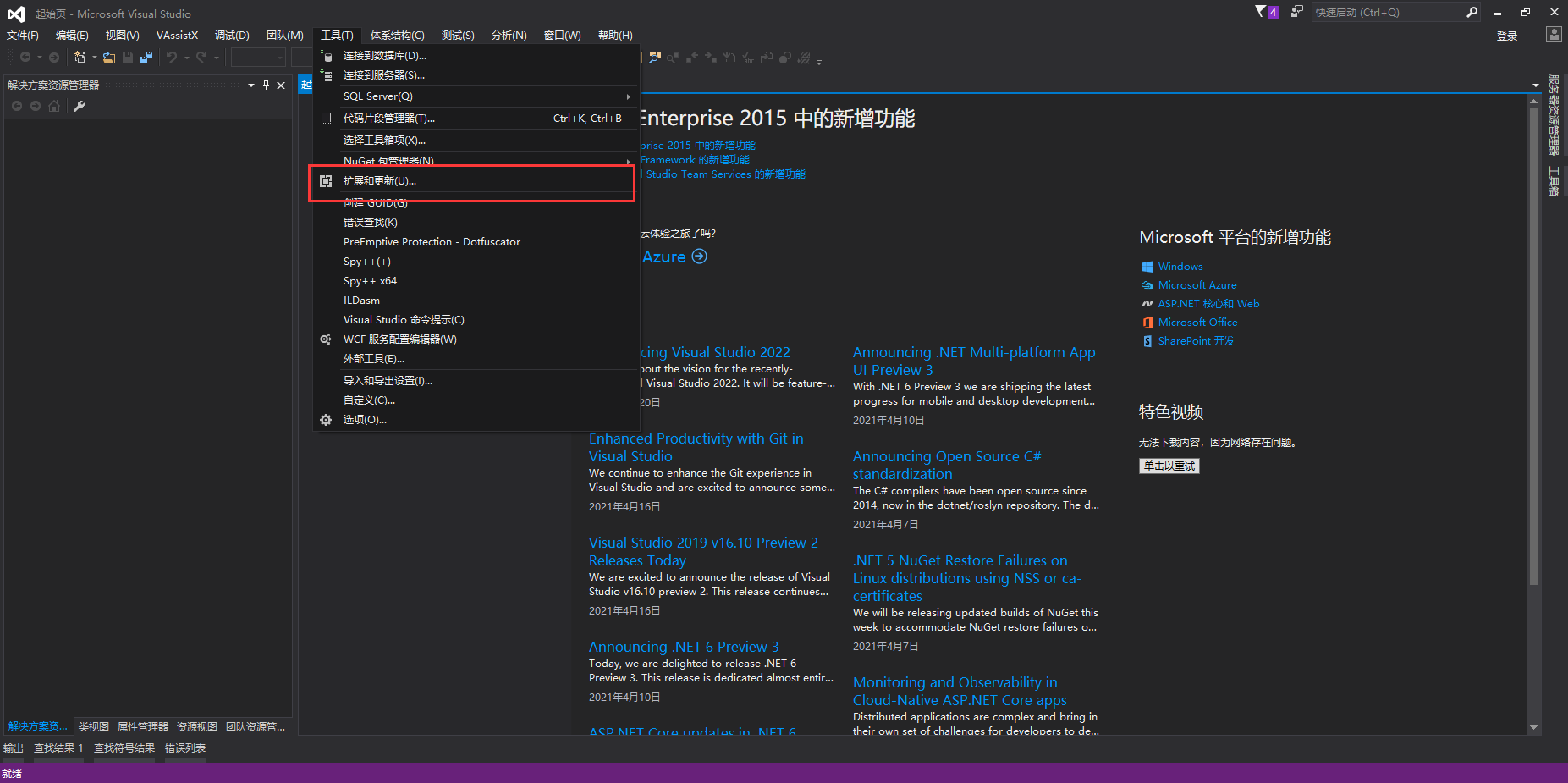
# 注释规范

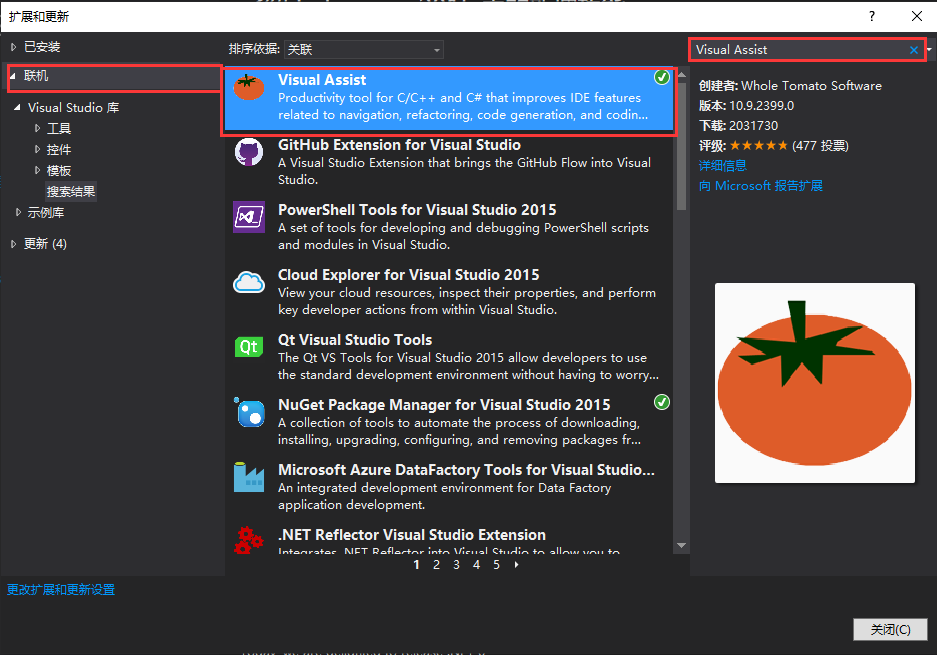
## VC注释工具 Visual Assist X

增强[Visual Studio](https://baike.baidu.com/item/Visual%20Studio" \t "_blank)开发环境下的编辑能力。Visual Assist X具有强大的编辑特色，提高了生产效率，可以完全集成到您的Microsoft开发环境中，升级了您的IDE，在不改变编程习惯的同时就可以感受到Visual Assist X为您带来的好处[1]  。

Visual Assist X是一款非常好的Microsoft Visual Studio[插件](https://baike.baidu.com/item/%E6%8F%92%E4%BB%B6)，可以支持Microsoft Visual Studio 2003，Microsoft Visual Studio 2005，Visual Studio 2008，Microsoft Visual Studio 2010，Microsoft Visual Studio 2013，Microsoft Visual Studio 2015，支持C/C++，[C#](https://baike.baidu.com/item/C%23" \t "_blank)，ASP，VisualBasic，[Java](https://baike.baidu.com/item/Java/85979" \t "_blank)和[HTML](https://baike.baidu.com/item/HTML" \t "_blank)等语言，能自动识别各种[关键字](https://baike.baidu.com/item/%E5%85%B3%E9%94%AE%E5%AD%97" \t "_blank)、[系统函数](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E5%87%BD%E6%95%B0)、[成员变量](https://baike.baidu.com/item/%E6%88%90%E5%91%98%E5%8F%98%E9%87%8F)、自动给出输入提示、自动更正大小写错误、自动标示错误等，有助于提高开发过程的自动化和开发效率。

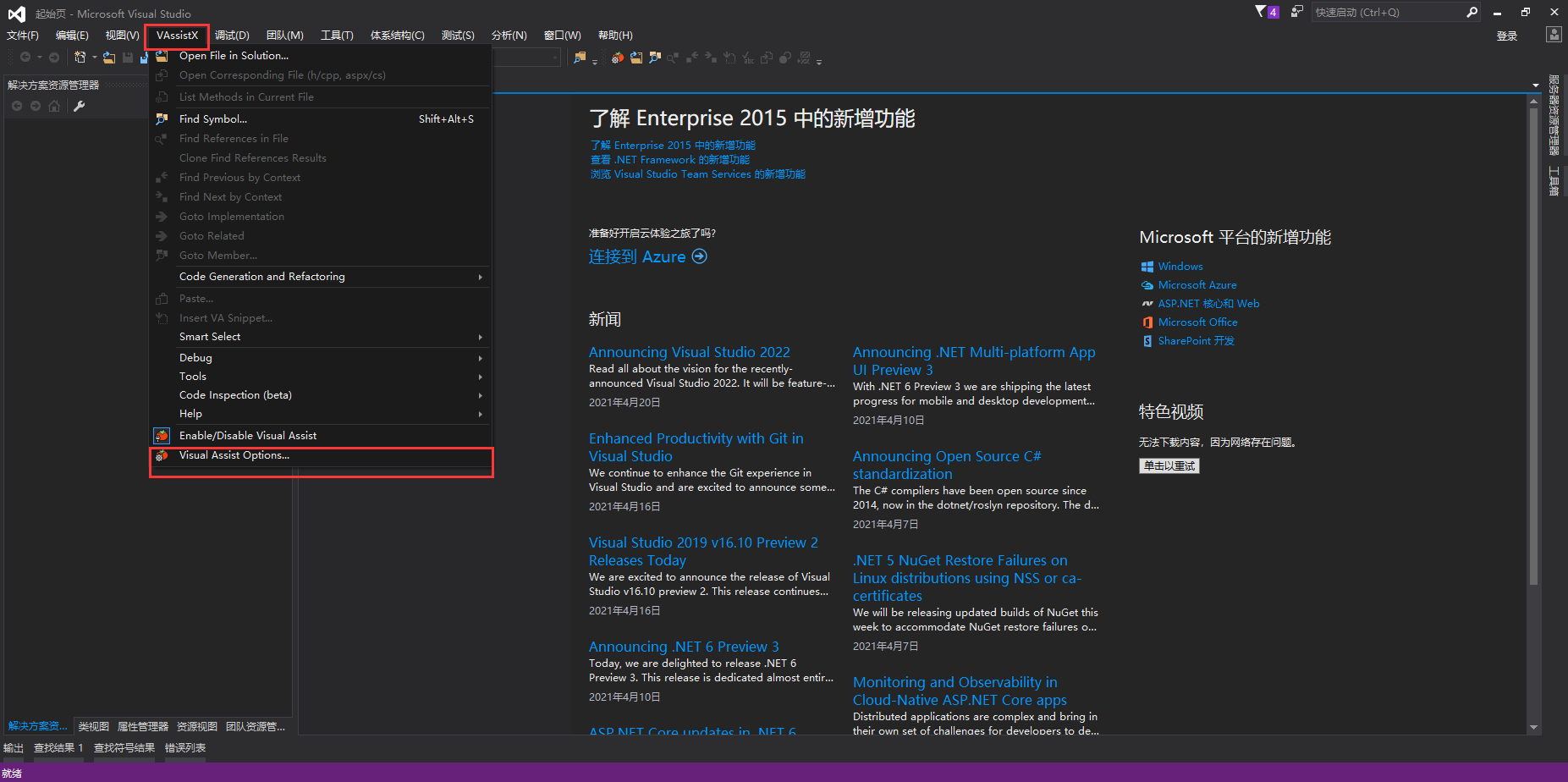
### Visual Assist X 安装

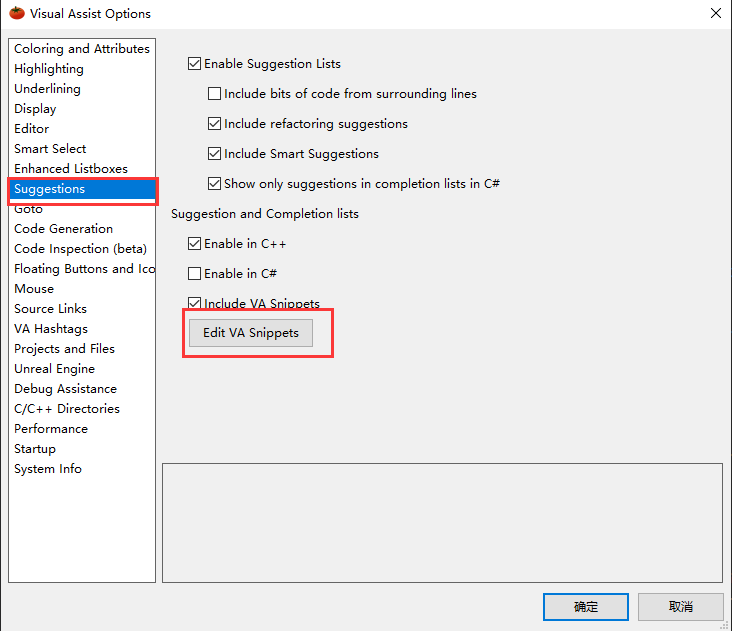




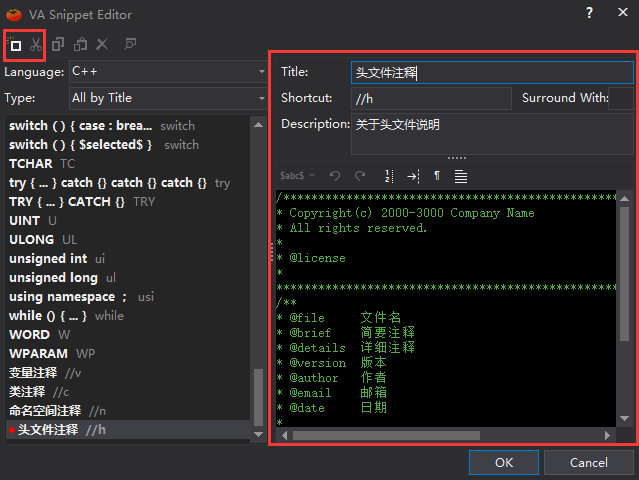
### Visual Assist X 快捷注释设置

安装好Visual Assist X后，打开VC开发工具。这里以VS2015为例。点击Visual Assist X Options，即工具栏上的番茄图标或菜单栏VassistX-Visual Assist X Options，会弹出设置对话框，点击Suggestions-Edit VA Snippets，如下图：





新建快捷编辑方案，填写标题、快捷方式、说明、注释样板，点击ok完成



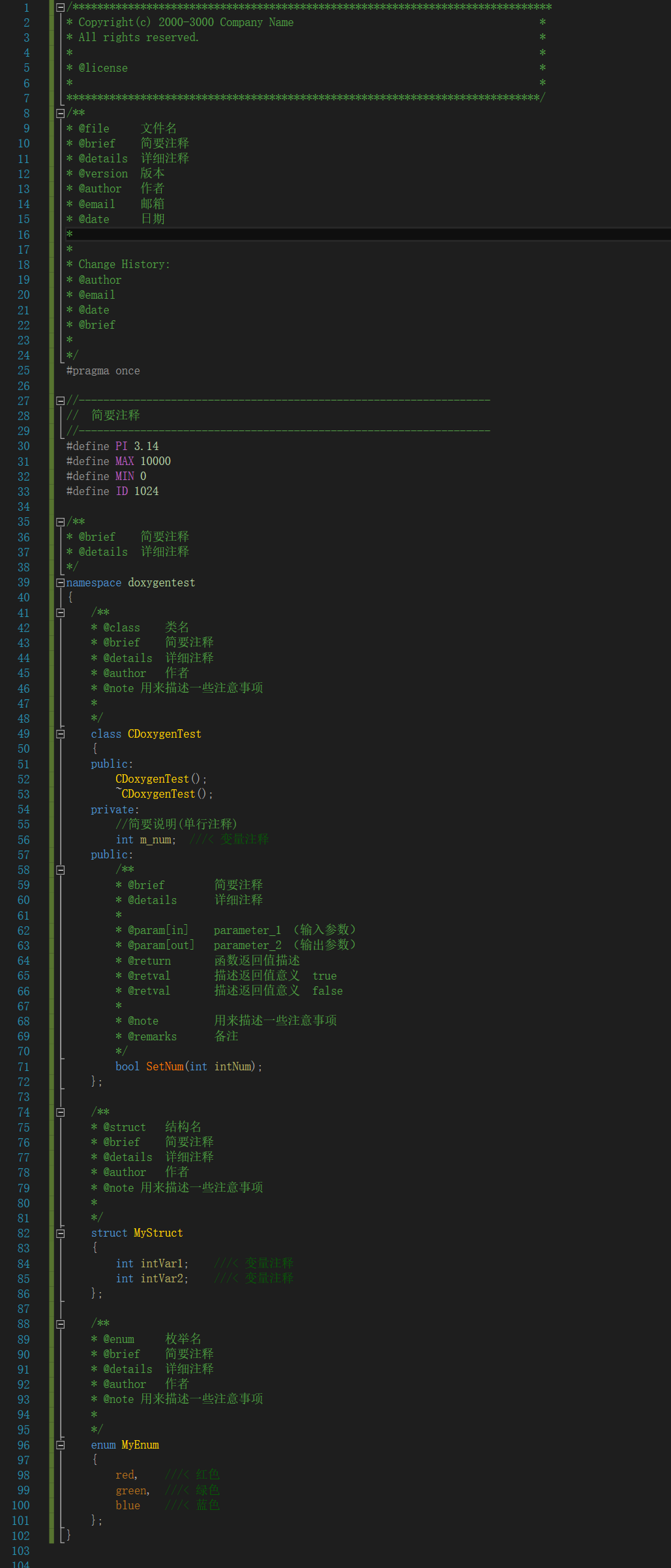
## 注释风格

程序代码注释风格将遵循Doxygen风格，Doxygen是一种开源跨平台的，以类似JavaDoc风格描述的文档系统，完全支持C、C++、Java、[Objective-C](http://baike.baidu.com/view/459423.htm" \t "_blank)和IDL语言，部分支持PHP、C#。注释的语法与Qt-Doc、KDoc和JavaDoc兼容。Doxgen可以从一套归档[源文件](http://baike.baidu.com/view/385166.htm" \t "_blank)开始，生成HTML格式的在线类[浏览器](http://baike.baidu.com/view/7718.htm" \t "_blank)，或离线的LATEX、RTF参考手册。

## 常用指令说明



## 代码注释示例



## 单行注释

需要注释代码的上一行，以//开头+注释类容



## 标注总叙

//-------------------------------------------------------------------

// 简要注释

//-------------------------------------------------------------------

## 头文件注释

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Copyright(c) 2000-3000 Company Name \*

\* All rights reserved. \*

\* \*

\* @license \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* @file 文件名

\* @brief 简要注释

\* @details 详细注释

\* @version 版本

\* @author 作者

\* @email 邮箱

\* @date 日期

\*

\*

\* Change History:

\* @author

\* @email

\* @date

\* @brief

\*

\*/

## 命名空间

/\*\*

\* @brief 简要注释

\* @details 详细注释

\*/

## 类、结构、枚举注释

### 类

/\*\*

\* @class 类名

\* @brief 简要注释

\* @details 详细注释

\* @author 作者

\* @note 用来描述一些注意事项

\*

\*/

### 结构

/\*\*

\* @struct 结构名

\* @brief 简要注释

\* @details 详细注释

\* @author 作者

\* @note 用来描述一些注意事项

\*

\*/

### 枚举

/\*\*

\* @struct 结构名

\* @brief 简要注释

\* @details 详细注释

\* @author 作者

\* @note 用来描述一些注意事项

\*

\*/

## 函数注释

/\*\*

\* @brief 简要注释

\* @details 详细注释

\*

\* @param[in] parameter\_1 （输入参数）

\* @param[out] parameter\_2 （输出参数）

\* @return 函数返回值描述

\* @retval 描述返回值意义 true

\* @retval 描述返回值意义 false

\*

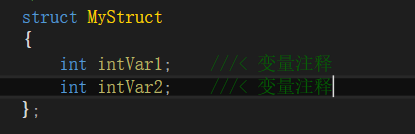
\* @note 用来描述一些注意事项

\* @remarks 备注

\*/

## 变量标注

变量名所在行后面+ ///< 变量注释



# 命名规范

## 公用规则

1. 只能是字母（A-Z，a-z）和数字（0-9）或者下划线（\_）组成，区分大小写。
2. 文件、函数、变量命名应具有描述性；
3. 类、自定义类型和变量命名应使用名词，函数名使用名词或动、名词结合的方式命名。
4. 除非表意十分清晰，否则不使用缩写。
5. 不加类型前缀，只保留全局变量、类成员变量和常量前缀。

## 文件命名

文件名全部使用小写，单词间以下划线（ \_ ）间隔。如：a\_file.cpp。

C++文件以.cpp结尾，头文件以.h结尾。

类或模块定义时文件名一般一一对应，如file.cpp和file.h。

## 类型命名

类、结构体、自定义类型和枚举类型均使用相同原则：每个单词都以大写字母开头，不包含下划线。

若集成MFC开发环境，也可遵循MFC开发标准，类名前加前缀C，例如：CMyDialog

## 宏和枚举命名

尽量不使用宏。

宏和枚举值均全部大写，单词间用下划线（ \_ ）间隔。如：MAX\_VALUE。

## 函数命名

函数第一个单词首字母大小写均可，后面单词都首字母大写，不包含下划线。

动词、名词结合，指明函数用途。如：GetNames()，getNames().

若继承自MFC类，类方法第一个单词首字母必须大写

## 命名空间

命名空间的名称是全小写的，其命名基于项目名称和目录结构：google\_awesome\_project

## 变量命名

变量名除第一个单词外，其他单词首字母大写。

类成员变量加前缀m\_ ，全局变量加前缀g\_，结构体成员变量与普通变量相同。

**特殊**：i、j、k作为循环变量，tmp作为临时变量。

## 常量命名

加前缀k，并且每个单词首字母大写。如：kMaxValue。