**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**об оформлении исходного кода программного обеспечения**

Часть 2.

Правила оформления исходного кода программного обеспечения с помощью команд Doxygen и html-тегов

Редакция 01

**История изменений**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| п/п | Дата | Номер версии | Статус | Автор | Причина изменения, № извещения | Комментарии |
| 1 | 02.04.2019 | 1 |  | Кислая М.В.,  Панкратов Д.Г. |  | Создание документа |
|  |  |  |  |  |  |  |

**Содержание**

[1 Введение 4](#_Toc14446447)

[1.1 Назначение и область применения методических указаний 4](#_Toc14446448)

[1.2 Ссылки 5](#_Toc14446449)

[1.3 Термины, определения и сокращения 5](#_Toc14446450)

[1.3.1 Термины и определения 5](#_Toc14446451)

[1.3.2 Сокращения 6](#_Toc14446452)

[2 Руководство по обеспечению соблюдения 7](#_Toc14446453)

[3 Общие правила 8](#_Toc14446454)

[3.1 Основные правила комментирования кода 8](#_Toc14446455)

[3.1.1 Краткое описание 9](#_Toc14446456)

[3.1.2 Подробное описание 10](#_Toc14446457)

[3.1.3 Параметры 11](#_Toc14446458)

[3.1.4 Возвращаемые значения 12](#_Toc14446459)

[3.1.5 Заметки 13](#_Toc14446460)

[3.2 Комментирование главной страницы проекта и описание модулей 14](#_Toc14446461)

[3.3 Комментирование заголовочного блока файла 18](#_Toc14446462)

[3.4 Комментирование истории изменений 21](#_Toc14446463)

[4 Комментирование элементов исходного кода 24](#_Toc14446464)

[4.1 Описание функций 24](#_Toc14446465)

[4.2 Описание макросов 26](#_Toc14446466)

[4.3 Описание переменных 28](#_Toc14446467)

[4.4 Описание определения типов и структуры данных 30](#_Toc14446468)

[4.5 Описание определения типов и перечислений 31](#_Toc14446469)

1. Введение

МУ состоит из двух частей:

1. Часть 1. Правила оформления исходного кода программного обеспечения на языке С;
2. Часть 2. Правила оформления исходного кода программного обеспечения с помощью команд Doxygen и html-тегов;
3. Часть 3. Правила написания исходного кода программного обеспечения на языке С.

Требования данных МУ являются обязательными для всех подразделений предприятия, принимающих участие в разработке кода ПО.

Данные МУ формулируют правила:

1. оформления исходного кода ПО на языке С;
2. комментирования кода с помощью системы документирования исходных текстов Doxygen;
3. оформления описания модулей с использованием html-тегов;
4. написания исходного кода ПО на языке С.
   1. Назначение и область применения методических указаний

Данные МУ устанавливают правила комментирования исходного кода с помощью команд Doxygen.

На сегодняшний день Doxygen фактически стал стандартом для документирования ПО.

Doxygen на основании исходного кода, содержащего команды вида \command, позволяет генерировать информативную документацию.

Для корректного комментирования кода с помощью Doxygen, в МУ рассмотрены общие принципы оформления комментариев, а также результат их применения на примере кода проекта ГКЛС-К.

Для описания модулей проекта используется язык гипертекстовой разметки HTML комплексно с таблицей каскадных стилей CSS.

Приведенные в МУ правила оформления кода имеют две степени строгости:

1. Обязательные (О) – правила, требующие обязательного выполнения и верификации. Невыполнение правил с такой степенью строгости недопустимо. Устранение несоответствий должно предполагать изменение исходного кода;
2. Рекомендуемые (Р) – правила, выполнение которых крайне желательно. Несоответствия, выявленные при верификации исходного кода, не влекут за собой обязательное изменение кода. Невыполнение правил с такой степенью строгости допустимо только при обоснованной аргументации (в случае использования ранее верифицированного кода).

Строгость правил оформления кода устанавливается таблицами (см. разделы 3-4), где для каждого правила определен специальный идентификатор (ДК – документирование кода) и указана степень строгости (О – обязательное; Р – рекомендуемое).

* 1. Ссылки

1. ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения;
2. Методические указания. Оформление исходного кода программного обеспечения. Часть 1. Правила оформления исходного кода программного обеспечения на языке С.
   1. Термины, определения и сокращения
      1. Термины и определения

Для применения данного документа имеют значение следующие термины и определения:

**Дочерний html-тег** – это html-тег, который входит в другой (родительский) html-теги

**Компиляция** – трансляция программы с языка высокого уровня в форму, близкую к программе, на машинном языке.

**Компоновка** – процесс объединения объектных файлов проекта и используемых библиотек в единую целевую сущность для целевой платформы. Это может быть исполняемая программа или библиотека статического или динамического типа.

**Родительский html-тег** – это html-тег, который включает в себя другие (дочерние) html-теги.

**Уровень вложенности** – это входжение дочернего html-тега в родительский html-тег.

* + 1. Сокращения

В данном документе используются следующие сокращения:

**ИС** – инструментальные средства.

**МУ** – методические указания.

**НТЦ** – научно-технический центр.

**ПО** – программное обеспечение.

1. Руководство по обеспечению соблюдения

Верификация исходного кода на языке C должна выполняться специалистами НТЦ, из состава отдела ПО.

Соблюдение МУ является обязательным. Исполнители (программисты) и их непосредственные руководители (начальник отдела, руководитель рабочей группы, ведущий по проекту) несут ответственность за выполнение правил.

Объектом верификации является исходный код на языке С, представленный в виде проекта – набора файлов, из которого при компиляции и компоновке получается исполняемый код.

Верификация правил должна выполняться путем анализа соответствия исходного кода, разработанного программистами отдела ПО, установленным в данных МУ правилам.

Верификация правил оформления включает следующие этапы:

1. анализ соответствия исходного кода определенным в данных МУ правилам;
2. документирование результатов верификации.

Верификация правил оформления выполняется с помощью ИС. Визуальный обзор выполняется путем анализа исходного кода с целью выявления несоответствий правилам. В процессе визуального обзора используются ИС MPLAB и Notepad++. Также список замечаний к коду можно посмотреть в файле warnings.log, который формируется после генерации файла справки Doxygen.

1. Общие правила
   1. Основные правила комментирования кода

Таблица 1 ‒ Требования к комментированию кода

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | При размещении комментария в одной строке с кодом, между кодом и комментарием Doxygen "///<" должно быть не менее 2 базовых отступов с выравниванием комментариев одного и того же типа элемента кода по отступу от самого длинного. | О |
|  | Команда # используется для создания гиперссылки на обьявление элемента кода. Ссылка на элемент кода создается путем добавления символа "#" непосредственно перед кодом. | О |
|  | Для создания в комментарии гиперссылки на модуль используется полное имя файла с указанием его расширения. | О |
|  | При комментировании используется следующий порядок команд Doxygen:   * \brief * \details * \param * \return или \retval * \note   Порядок команд остается неизменным даже при отсутствии одной или больше команд. | О |
|  | Блок комментариев Doxygen и код должны разделяться строкой, которая состоит только из символов "///". | О |
|  | Для создания маркированного списка (знак маркера – черная точка), текст названия списка должен записываться с прописной буквы через 1 пробел от команды. | О |
|  | Для создания элемента списка необходимо с новой строки через 5 пробелов от "///" поставить символ "–". Текст описания элемента списка должен записываться со строчной буквы через 1 пробел от символа "–". При этом в конце описания последнего элемента списка ставится символ ".", а в конце остальных – символ ";". | О |
|  | Список и дальнейшее описание (при его наличии) должны разделяться строкой, которая состоит только из символов "///". | О |
|  | При использовании спецификатора extern описание командами Doxygen является обязательным, а при объявлении элемента кода дублируется только краткое описание обычным комментарием языка С. | О |

Таблица 2 – Шаблон оформления списка

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | 120 символов  ///‧\command‧Название списка:  ///‧‧‧‧‧–‧первый пункт списка;  ///‧‧‧‧‧–‧последний пункт списка.  ///  /// \command Описание. |

* + 1. Краткое описание

Таблица 5 ‒ Требования к оформлению подробного описания

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Для указания краткого описания элемента кода используется команда \brief. | О |
|  | Описание должно записываться с прописной буквы через 1 пробел от команды \brief и состоять из 1 предложения с символом "." в конце описания. | О |
|  | Для указания краткого описания элемента используется обычный комментарий Doxygen "///<", который с прописной буквы записывается через 1 пробел от "///<" с символом "." в конце описания. Размещается в той же строке что и код. | О |

Таблица 4 – Шаблоны оформления комментариев краткого описания

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | ///‧\brief‧Краткое описание элемента кода. |
| 2 | void ИмяФункции( void );‧‧‧‧‧‧‧‧///<‧Краткое описание элемента кода. |

* + 1. Подробное описание

Таблица 5 ‒ Требования к оформлению подробного описания

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Для указания подробного описания элемента кода используется команда \details. | О |
|  | Подробное описание должно записываться с прописной буквы через 1 пробел от команды \details с символом "." в конце описания. | О |
|  | Для размещения подробного описания в 2-х и больше строках, достаточно указать команду \details только в первой строке. Следующие строки подробного описания должны начинаться с символов "///". | О |
|  | Если текст подробного описания касается разных аспектов, то команда \details указывается каждый раз и пишется с новой строки. | О |

Таблица 6 – Шаблоны оформления комментариев подробного описания

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | ///‧\details‧Подробное описание элемента кода. |
| 2 | 120 символов  ///‧\details‧Подробное описание элемента кода.  ///‧Продолжение подробного описания элемента кода. |
| 3 | 120 символов  ///‧\details‧Первое подробное описание элемента кода.  ///‧\details‧Второе подробное описание элемента кода. |

* + 1. Параметры

Таблица 7 ‒ Требования к оформлению параметров

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Для указания параметров элемента кода используется команда \param. Через 1 пробел после команды должен записываться параметр, после которого через 1 пробел должен ставиться символ "–". | О |
|  | Описание параметра должно записываться со строчной буквы через 1 пробел от символа "–" и состоять из 1 предложения. | О |
|  | Для 2-х и больше параметров команда указывается каждый раз и пишется с новой строки. При этом в конце описания последнего параметра ставится символ ".", а в конце остальных – символ ";". | О |
|  | Если параметр – указатель, то символ "\*" не пишется, но обязательно записывается слово "указатель". | О |

Таблица 8 – Шаблоны оформления комментариев параметра

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | ///‧\param‧параметр1‧-‧параметр элемента кода. |
| 2 | ///‧\param‧параметр1‧-‧параметр элемента кода;  ///‧\param‧параметр2‧-‧параметр элемента кода. |

* + 1. Возвращаемые значения

Таблица 9 ‒ Требования к оформлению возвращаемых значений

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Для указания возвращаемых значений используется команда \return. | О |
|  | Описание возвращаемого значения с помощью команды \return должно записываться с прописной буквы через 1 пробел от команды и состоять из 1 предложения с символом "**.**" в конце описания. Причем слово "возвращает" не пишется. | О |
|  | Для указания возвращаемых значений логического типа данных используется команда \retval. | О |
|  | Тип значения (true, false) должен записываться строчными буквами через 1 пробел от команды \retval. | О |
|  | После типа значения true символ "–" должен записываться через 2 пробела, а после false – через 1 пробел. | О |
|  | Описание возвращаемого значения должно записываться со строчной буквы через 1 пробел от символа "–" и состоять из 1 предложения. В конце первой строки ставится символ ";", в конце второй – символ ".". | О |

Таблица 10 – Шаблоны оформления комментариев возвращаемых значений

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | ///‧\return‧Возвращаемое значение. |
| 2 | ///‧\retval‧true‧‧–‧возвращаемое значение1;  ///‧\retval‧false‧–‧возвращаемое значение2. |

* + 1. Заметки

Описание заметки должно содержать информацию о вызове элемента кода, времени вызова, его связи с другими элементами и т.д.

Таблица 11 ‒ Требования к оформлению заметок

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Для указания заметки используется команда \note. | О |
|  | Текст заметки должен записываться с прописной буквы через 1 пробел от команды \note с символом "." в конце описания. | О |
|  | Для размещения заметки в 2-х и больше строках, достаточно указать команду \note только в первой строке. Следующие строки заметки должны начинаться с символов "///". | О |
|  | Если текст заметки касается разных аспектов, то команда \note указывается каждый раз и пишется с новой строки. | О |

Таблица 12 – Шаблоны оформления комментариев заметки

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | ///‧\note‧Текст заметки. |
| 2 | 120 символов  ///‧\note‧Текст заметки.  ///‧Продолжение текста заметки. |
| 3 | 120 символов  ///‧\note‧Текст первой заметки.  ///‧\note‧Текст второй заметки. |

* 1. Комментирование главной страницы проекта и описание модулей

Таблица 13 ‒ Требования к оформлению главной страницы проекта и описания модулей

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Главная страница проекта должна иметь название Description с расширением "txt" (маска для файлов – \*.txt). и подключаться к Doxygen относительным путем файла во вкладке Expert пункт Input блок INPUT. | О |
|  | Описание главной страницы проекта и каждого файла должно начинаться с символов "/\*\*". | О |
|  | Название страницы (main\_page) должно указываться с новой строки через 3 пробела после команды \anchor. | О |
| \anchor main\_page | | |
|  | Название документа (в соответствии с МУ о порядке обозначения программ и ПД) должно указываться с большой буквы с новой строки через 1 пробел после команды \mainpage без точки в конце названия. | О |
| \mainpage Название документа | | |
|  | При написании кода должна использоваться вложенность тегов, величина отступа дочернего тега относительно родительского – 2 пробела. | О |
|  | Название раздела главной страницы и описания модуля должно указываться с помощью парного тега блочного элемента <div>…</div> и универсального атрибута class, который имеет значение "sectionname", указанное через символ "=" без дополнительных пробелов. | О |
| <div class="sectionname">Название раздела</div> | | |
|  | Главная страница проекта должна содержать следующие разделы:   * Введение * Спецификация файлов * 1. Краткое описание изделия * 2. Термины, определения и сокращения * 3. Функциональные требования к ПО * 4. Архитектура ПО * 5. Требования к реализации ПО | О |
|  | Текст раздела главной страницы и описания модуля должен указываться с помощью парного тега блочного элемента <p>…</p> и универсального атрибута class, который имеет значение "text", указанное через символ "=" без дополнительных пробелов. При этом текст раздела указывается на следующей строке под открывающимся тегом. Закрывающийся тег должен записываться в конце последней строки текста раздела, без пробела перед ним. | О |
| <p class="text">  Текст раздела.</p> | | |
|  | Ссылка на файл должна указываться с помощью парного тега <a>…</a> и атрибута href, который задает адрес документа, указанный через символ "=" без дополнительных пробелов. | О |
| <a href="Имя\_файла.ext">Гиперссылка</a> | | |
|  | Абзац в тексте раздела должен указываться с помощью тега <br> в конце первого абзаца. Следующий абзац раздела указывается с новой строки без дополнительных отступов. | О |
| Текст первого абзаца.<br>  Текст второго абзаца. | | |
|  | Описание главной страницы и файла должно заканчиваться символами "\*/". | О |
|  | Описание модуля должно содержаться в файле "Description.txt" с одной пустой строкой после последнего описания. | О |
|  | Название страницы с описанием модуля должно указываться с большой буквы с новой строки через 2 пробела после команды \page без точки в конце описания. | О |
| \page Название страницы с описанием модуля | | |
|  | Краткое описание модуля указывается через 1 пробел с большой буквы с помощью команды \brief с точкой в конце описания. | О |
| \brief Краткое описание модуля. | | |
|  | Для создания нумерованного списка используется парный тег <ol>…</ol>, для указания элементов списка используется парный тег <li>…</li>. При этом в конце описания последнего элемента списка ставится символ ".", а в конце остальных – символ ";". | О |
| <ol>  <li>первый элемент нумерованного списка;</li>  <li>последний элемент нумерованного списка.</li>  </ol> | | |
|  | Для создания маркированного списка используется парный тег <ul>…</ul>, для указания элементов списка используется парный тег <li>…</li>. При этом в конце описания последнего элемента списка ставится символ ".", а в конце остальных – символ ";". | О |
| <ul>  <li>первый элемент нумерованного списка;</li>  <li>последний элемент нумерованного списка.</li>  </ul> | | |
|  | Для написания формул используется библиотека для отображения математических формул MathJAX, для отображения нужно включить в Doxygen в вкладке Expert пункта HTML блок USE\_MATHJAX. | О |
|  | Для вставки рисунка используется парный тег блочного элемента <p>…</p> и универсальный атрибут class, который имеет значение "images", указанное через символ "=" без дополнительных пробелов. Идентификатор рисунка должен указываться на новой строке с помощью парного тега блочного элемента <div>…</div> и атрибута id, который имеет значение "imgx", указанное через символ "=" без дополнительных пробелов. Для вставки рисунка необходимо использовать команду @image html. Через 1 пробел указывается имя рисунка с его расширением (рисунок должен содержаться в папке DOC/IMAGE). | О |
|  | Название рисунка должно указываться с новой строки и выравнивается по центру с помощью парного тега <center>…</center>. | О |
| <p class ="images">  <div id="imgx">  @image html Имя\_рисунка.jpg</div>  <center>Рисунок x – Название\_рисунка</center></p> | | |
|  | Ссылка на рисунок должна указываться с помощью парного тега <a>…</a> и атрибута href, который задает якорь рисунка "#img1", "#img2", .., указанный через символ "=" без дополнительных пробелов. Должен указываться универсальный атрибут class, который имеет значение "objlink", указанное через символ "=" без дополнительных пробелов. | О |
| <a href="#imgx" class="objlink">рисунке х</a> | | |

* 1. Комментирование заголовочного блока файла

Таблица 14 ‒ Требования к оформлению блока комментариев, обобщенно описывающего содержание файла, его авторов и версию

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Описание заголовочного блока файла должно начинаться с символов "/\*\*". | О |
|  | Каждая строка заголовочного блока файла должна начинаться с символа "\*", а команда записывается через 1 пробел. | О |
|  | Для отображения комментариев заголовочного блока файла используется команда \file. Имя файла с расширением указывается через 4 пробела от команды \file без символа "." в конце описания. | О |
|  | Для указания краткого описания назначения модуля используется команда \brief. Краткое описание заголовочных файлов должно записываться с прописной буквы через 3 пробела от команды \brief и состоять из 1 предложения без символа "." в конце описания. | О |
|  | Краткое описание файлов с исходным кодом должно записываться через 3 пробела от команды \brief и состоять из команды \copybrief с указанием через 1 пробел имени соответствующего заголовочного файла с указанием его расширения без символа "." в конце описания. | О |
|  | Для указания подробного описания назначения модуля используется команда \details. Краткое описание заголовочных файлов должно дополняться подробным описанием, которое должно записываться через 1 пробел от команды \details и состоять из слова «Интерфейс» без символа "." в конце описания. | О[[1]](#footnote-1) |
|  | Подробное описание должно записываться с прописной буквы через 1 пробел от команды \details и содержать следующий текст «Принцип работы и использование компонента описан на странице \ref Описание\_модуля», где \ref – команда, которая используется для создания ссылки на страницу (\page) Описание\_модуля файла "Description.txt". | О |
|  | При достижении ограничения по количеству символов в строке, подробное описание переносится на следующую строку, которая начинается с символа "\*". | О |
|  | Строка после подробного описания должна состоять только из символа "\*". | О |
|  | Для указания версии файла используется команда \version(формат Х.Х.Х)[[2]](#footnote-2). Номер версии должен совпадать с последним номером версии из истории изменений и записываться через 1 пробел от команды \versio**n**. | О |
|  | Для указания даты создания или последнего изменения файла используется команда \date(формат ДД-ММ-ГГГГ). Дата должна совпадать с датой последней версии из истории изменений и записываться через базовый отступ от команды \date. | О |
|  | Для указания автора файла используется команда \author. Фамилия и инициалы автора (формат Фамилия И.О.) должны записываться через 2 пробела от команды \author. | О |
|  | Фамилия и инициалы автора и корректировщика файла в строке команды \authorдолжны записываться через символ "**,**" и 1 пробел. | О |
|  | Описание заголовочного блока файла должно заканчиваться символами "\*/" с новой строки. | О |
|  | Команды \brief и \details в заголовочном блоке файлов \*.h и \*.c должны иметь одинаковые комментарии для одного и того же модуля. | О |
|  | В краткое описание файла \*.h, оформленное с помощью команды \brief, через 1 пробел дописывается слово "Интерфейс" с символом "." в конце. Исключением является наличие одного файла модуля (в этом случае, слово "Интерфейс" не дописывается). | О |
| *Measurement.h*  *Вид файла справки Doxygen:* | | |

Таблица 15 – Шаблоны оформления заголовочного блока файла

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | Файл \*.h |
| /\*\*  \*‧\file Имя\_файла.h  \*‧\brief‧‧‧Краткое описание  \*‧\details Интерфейс  \* \details Принцип работы и использование компонента описан на странице \ref Имя\_файла.html  \*‧\details‧Подробное описание.  \*  \*‧\version‧Х.Х.Х  \*‧\date‧‧‧‧ДД-ММ-ГГГГ  \*‧\author‧‧Фамилия И.О.  \*/ |
| 2 | Файл \*.с |
| /\*\*  \*‧\file Имя\_файла.c  \*‧\brief‧‧‧\copybrief Имя\_файла.h  \* \details Принцип работы и использование компонента описан на странице \ref Имя\_файла.html  \*  \*‧\version‧Х.Х.Х  \*‧\date‧‧‧‧ДД-ММ-ГГГГ  \*‧\author‧‧Фамилия И.О.  \*/ |

* 1. Комментирование истории изменений

Таблица 16 ‒ Требования к оформлению истории изменений

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Описание истории изменений файла должно присутствовать в каждом файле и начинаться с символов "/\*\*". | О |
|  | Каждая строка истории изменений файла должна начинаться с символа "\*", а текст записываться через 1 пробел. | О |
|  | Пункт "История изменений:" должен записываться с прописной буквы с новой строки. Следующая строка должна состоять только из символа "\*". | О |
|  | Пункт "Версия" должен записываться с прописной буквы с новой строки. Через 1 пробел указывается версия файла (формат Х.Х.Х). | О |
|  | Нумерация версий начинается с 1.0.1. | О |
|  | Пункт "Дата" должен записываться с прописной буквы с новой строки. Через 3 пробела указывается дата создания или последнего изменения файла (формат ДД-ММ-ГГГГ). | О |
|  | Пункт "Автор" должен записываться с прописной буквы с новой строки. Через 2 пробела указывается фамилия и инициалы автора или корректировщика файла (формат Фамилия И.О.). Следующая строка должна состоять только из символа "\*". | О |
|  | Пункт "Описание версии:" должен записываться с прописной буквы с новой строки. | О |
|  | Текст описания версии должен записываться с прописной буквы через базовый отступ и содержать информацию об изменении кода относительно предыдущей версии. | О |
|  | Если номер версии 1.0.1, то текст описания – "Базовая версия". | О |
|  | Для разделения версий используется строка, которая состоит только из символа "\*". | О |
|  | Описание истории изменений должно заканчиваться символами "\*/". | О |

Таблица 17 – Шаблон оформления истории изменений

| **№** | **Шаблон оформления** |
| --- | --- |
| 1 | /\*\*  \*‧История‧изменений:  \*  \*‧Версия‧1.0.1  \*‧Дата‧‧‧ДД-ММ-ГГГГ  \*‧Автор‧‧Фамилия И.О.  \*  \*‧Описание‧версии:  \*‧‧‧‧Базовая версия.  \*  \*‧Версия‧Х.Х.Х  \*‧Дата‧‧‧ДД-ММ-ГГГГ  \*‧Автор‧‧Фамилия И.О.  \*  \*‧Описание‧версии:  \*‧‧‧‧Текст описания.  \*/ |

1. Комментирование элементов исходного кода

Doxygen автоматически выделяет следующие категории элементов:

1. функции;
2. макросы;
3. переменные;
4. определения типов;
5. структуры данных;
6. перечисления.

Правила оформления команд для каждой из категорий элементов Doxygen рассмотрены на примере кода проекта ГКЛС-К.

* 1. Описание функций

Таблица 18 ‒ Требования к оформлению функций

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | При инициализации описание элемента кода командами Doxygen является обязательным, а при реализации элемента кода дублируется только краткое описание обычным комментарием языка С. | О |
|  | Дополнительная информация (**\param, \return, \note**) указывается перед инициализацией функции после краткого описания, заданного командой **\brief**, и подробного описания, заданного командой **\details** (c сохранением порядка команд). | О |
| *DigitalFilter.h*  *DigitalFilter.с*  *Вид файла справки Doxygen:* | | |

* 1. Описание макросов

Таблица 19 ‒ Требования к оформлению макросов

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Дополнительная информация (**\details,** **\param, \return, \note**) указывается перед макросом (макросом типа функция) после краткого описания, заданного командой **\brief** (c сохранением порядка команд). | О |
| *asserts.h*  *Вид файла справки Doxygen:* | | |
|  | При наличии только краткого описания, оно должно записываться с прописной буквы обычным комментарием Doxygen "**///<**" с символом "**.**" в конце описания в той же строке что и макрос. | О |
| *Buttons.h*  *Вид файла справки Doxygen:* | | |
|  | При наличии в макросе типа функция цикла **do…while**, краткое описание записывается с помощью команды **\brief** с прописной буквы с символом "**.**" в конце описания. | О |
| *InterChannel.c*  *Вид файла справки Doxygen:* | | |

* 1. Описание переменных

Таблица 20 ‒ Требования к оформлению переменных

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | При наличии только краткого описания, оно должно записываться с прописной буквы обычным комментарием Doxygen "**///<**" с символом "**.**" в конце описания в той же строке что и переменная. | О |
| *SignalCPB.c*  *Вид файла справки Doxygen:* | | |
|  | Дополнительная информация (**\details**, **\note**) указывается перед переменной после краткого описания, заданного командой **\brief** (c сохранением порядка команд). | О |
| *ProtectionStateCodes.c*  *Вид файла справки Doxygen:* | | |

* 1. Описание определения типов и структуры данных

Таблица 21 ‒ Требования к оформлению определения типов и структуры данных

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Структура должна иметь краткое описание (оформленное с использованием команды **\brief**), а элементы структуры должны описываться с прописной буквы обычными комментариями Doxygen "**///<**" и символом "**.**" в конце описания. | О |
| *BinIn.h*  *Вид файла справки Doxygen:* | | |

* 1. Описание определения типов и перечислений

Таблица 22 ‒ Требования к оформлению определения типов и перечислений

| **Номер правила** | **Правило** | **Степень строгости** |
| --- | --- | --- |
|  | Дополнительная информация (**\details**, **\note**) указывается перед определением типа после краткого описания, заданного командой **\brief** (c сохранением порядка команд). Элементы перечислений должны описываться со строчной буквы обычными комментариями Doxygen "**///<**" без символа "." в конце описания. | О |
| *SignalCpbTesting.h*  *Вид файла справки Doxygen:* | | |

1. Правило применяется для заголовочного файла с соответствующим файлом с исходным кодом. [↑](#footnote-ref-1)
2. 1 цифра – изменяется при потере совместимости ПО и аппаратуры.

   2 цифра – изменяется при расширении функционала, добавлении новых интерфейсов, когда сохраняется совместимость.

   3 цифра – незначительное изменение кода.

   Обнуление всех младших цифр происходит при изменении старшей. [↑](#footnote-ref-2)