инистерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Высшая школа кибернетики и цифровых технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

ПО КУРСУ «ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

«ДОКУМЕНТИРОВАНИЕ КОДА»

Выполнил: студент 2 курса группы ПО(б) – 31

Астафьев Алексей Алексеевич

Проверила: ассистент ВШ КЦТ

Балабасова Алиса Дмитриевна

Хабаровск 2025 г.

Цель: освоить работу с инструментами автоматической генерации документации (Doxygen).

Задания:

1. Добавить комментарии в код по стандарту Doxygen (или другой на выбор).

2. Сгенерировать HTML-документацию.

3. Настроить автоматическое обновление документации при коммите (скриншоты).

Ход работы:

1. Добавить комментарии в код по стандарту Doxygen (или другой на выбор).

Для добавления комментариев в код по стандарту Doxygen ничего устанавливать не нужно, только необходимо посмотреть нужный синтаксис. Однако, для последующей генерации HTML-документации, скачаю Doxygen с официального сайта.

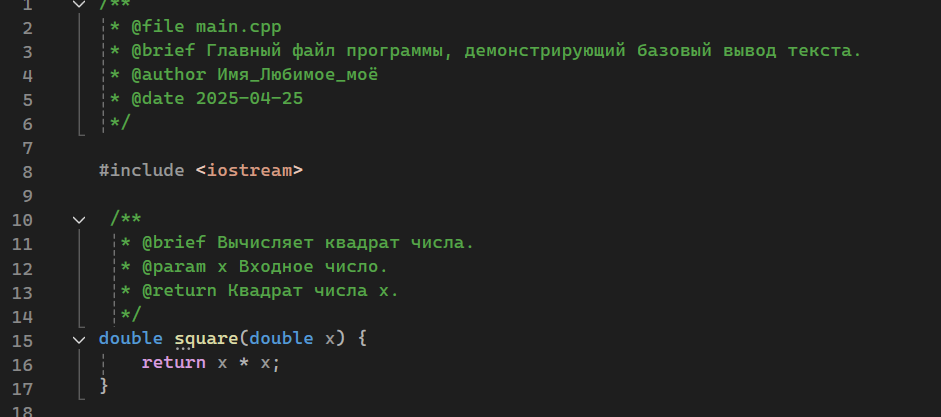


Рисунок 1 – добавленные комментарии по стандарту Doxygen

Doxygen использует специальные комментарии с тегами, начинающимися с @ или \. Например:

@file — описание файла.

@brief — краткое описание функции или класса.

@param — описание параметров функции.

@return — описание возвращаемого значения.

Комментарии могут быть в формате /\*\* ... \*/ (для многострочных) или /// (для однострочных).

2. Сгенерировать HTML-документацию.

Для генерации HTML-документации генерируется файл Doxygen, после чего его можно открыть в любом текстовом редакторе для настройки генерации документации: язык, название папки, где будет генерироваться документация, «проход» по всем папкам проекта (рекурсивность) и так далее.

В частности, я сменил язык генерации с английского на русский, так как комментарии у меня, очевидно, на русском.

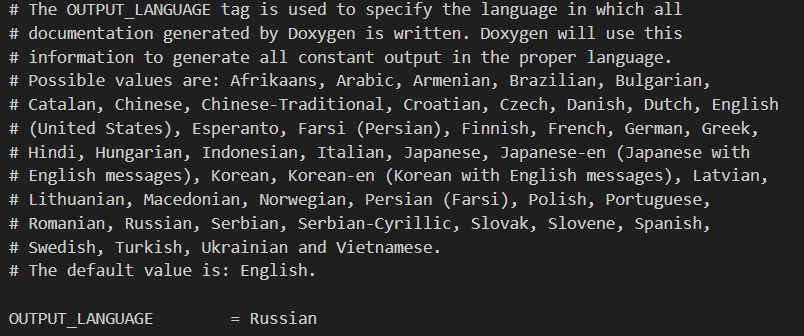


Рисунок 2 – файл Doxygen

После настройки генерации, пишем команду в консоли, и HTML-документация автоматически собирается.

3. Настроить автоматическое обновление документации при коммите (скриншоты).

Для автоматического обновления документации при коммите, создадим файл doxygen.yml, который будет автоматически обновлять документацию каждый раз, когда будут внесены изменения в проект.

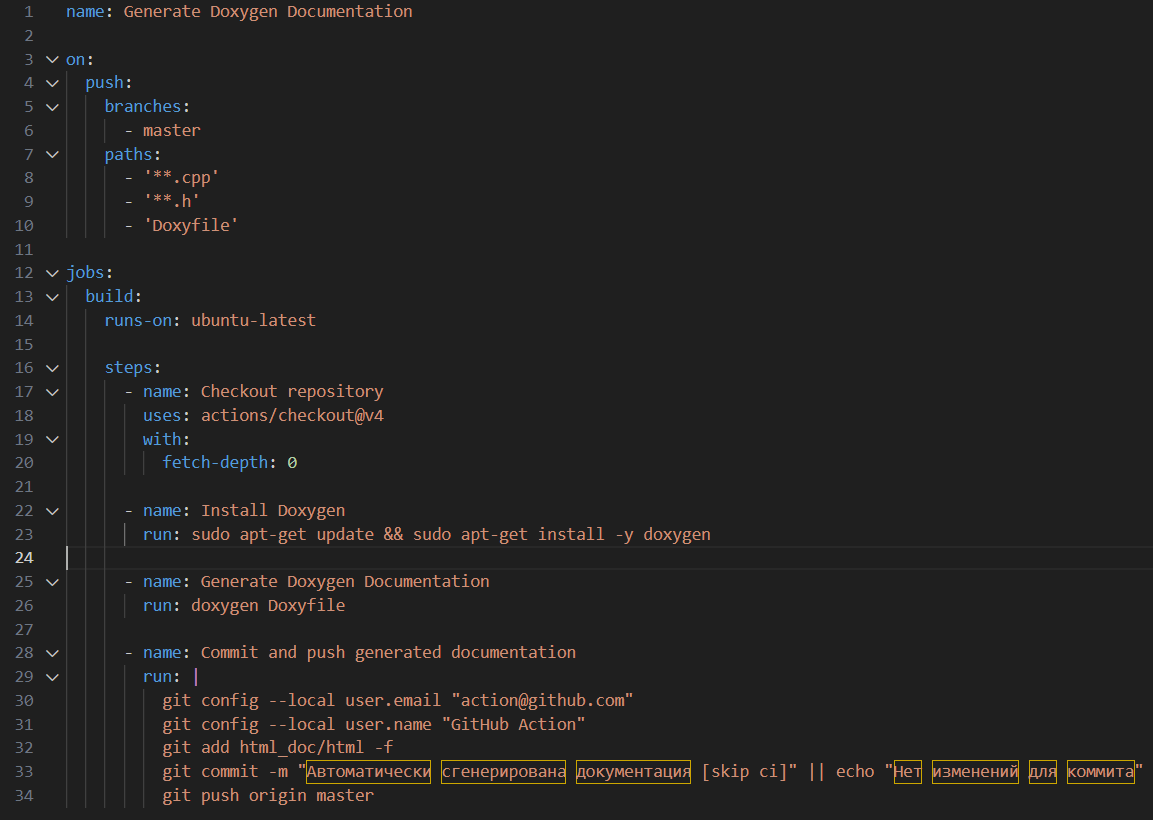


Рисунок 3 – файл doxygen.yml

paths: Ограничивает запуск workflow, чтобы он срабатывал только при изменении определённых файлов:

\*\*.cpp: Любые файлы с расширением .cpp (включая поддиректории, благодаря \*\*).

run: |: Многострочная команда, выполняемая в терминале.

Разбор команды:

git config --local user.email "action@github.com": Устанавливает email для коммита. Git требует, чтобы у каждого коммита был автор.

git commit -m "Автоматически сгенерирована документация [skip ci]" || echo "Нет изменений для коммита": Создаёт коммит с сообщением.

[skip ci] в сообщении коммита — это инструкция для GitHub Actions, чтобы не запускать workflow для этого коммита..

git push origin master: Пушит изменения в ветку master репозитория.

После изменения в файле main.cpp - добавления в него дополнительной функции и комментариев, workflow-run успешно выполнился. Автоматическое обновление документации прошло успешно.

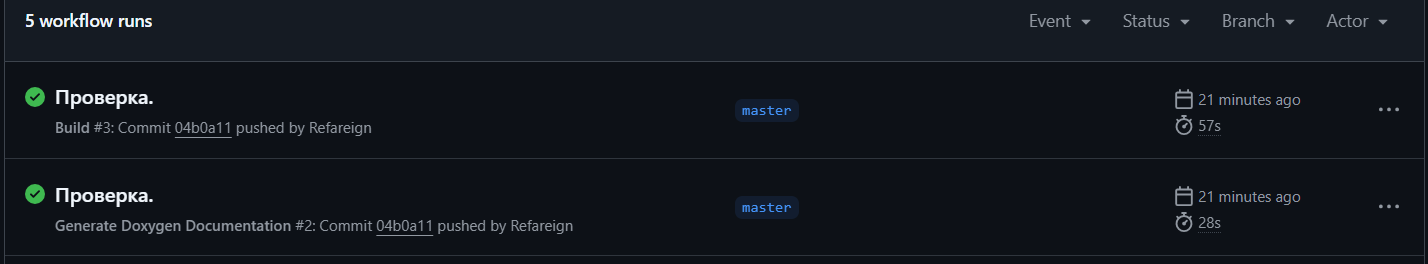


Рисунок 4 – успешное обновление документации.